

NISEI Comprehensive Rules Document

This rules document is to be used as reference material. It is not intended to be read straight through. If you still have questions after consulting this document, please ask us online via [email](#) or [Twitter](#).

This is **version 1.5** of this document, effective **13 August 2021**.

Summary of Changes

Changes from version 1.4 are **highlighted in orange**. See the System Gateway/System Update 2021 Release Notes for detailed explanations of the major changes.

- Clarified that *System Gateway Starter Pack* identities are not legal for deckbuilding ([rule 1.4.1a](#)).
- Clarified rules about when cards become active ([section 1.8.3](#)).
- Changed most uses of “token” to “counter”. A future update will fully deprecate the use of “token”, but at this time both words still appear on card text and are considered synonymous. See [section 1.9](#).
- Clarified what counts as having “placed” counters ([rule 1.9.2b](#)).
- Updated the rule about spending credits to better explain spending credits from cards ([rule 1.10.3c](#)).
- Updated rules about recurring credits to explain the timing more precisely ([section 1.10.5](#)).
- Added clarifying rules about gaining, losing, and spending clicks ([section 1.11.3](#)).
- Added the notion of “objects” to consolidate rules about how effects interact with cards moving around. ([section 1.12](#)). An object is a game construct that refers to a card or counter, but only until it moves to another zone. Abilities generally act on specific objects rather than the physical game pieces they correspond to.
- Cleaned up some language about targets and added rules for choosing subroutines as targets ([section 1.15](#)).
- Added an example to the rule about mandatory abilities making a cost unpayable ([rule 1.16.1b](#)).
- Clarified the rule about costs of 0 ([rule 1.16.1d](#)).
- Added a rule clarifying how additional costs interact with effects that ignore costs ([rule 1.16.9c](#)).
- Clarified that cards cannot be advanced by default ([rule 1.18.3](#)).
- Promoted the last sentence of [section 1.21.2](#) (“A player can look at facedown cards they control at any time.”) to its own subrule.
- Moved the rule for unique (◆) cards from [section 2](#) to [rule 10.1.1](#).
- Removed **connection** from the list of upgrade subtypes in [section 2.16.7](#), as *Corporate Troubleshooter* has now received errata to remove its subtype.

- Changed the rules for **consoles** to work like unique cards ([section 3.8.5](#)).
- Clarified the rule about how memory costs work ([section 3.9.3](#)).
- Updated rules for icebreakers ([section 3.9.5](#)) and ice encounters ([section 6.5](#)) to support the new “interface” keyword.
- Updated [rule 4.1.2a](#) to better highlight the exception for installing cards.
- Clarified the rule defining public zones ([rule 4.1.4](#)).
- Added a rule to highlight that a player’s deck is an ordered zone ([rule 4.2.3](#)).
- Expanded the zone and location rules to include counters. The bank is now defined as a zone ([section 4.7](#)), and the locations of counters in the play area are explained in [section 4.6.5](#).
- All servers now have roots. Installing cards in the root of a remote server works the same as installing cards “in” a remote server used to, but now terminology about where cards are installed is consistent across all servers. Reorganized rules about servers ([section 4.6.6](#)), central servers ([section 4.6.7](#)), and remote servers ([section 4.6.8](#)) to implement this change.
- Added [section 5.1.4](#) to clarify “turn begins” and “turn ends” abilities. In particular, “discard phase ends” is now used in some card abilities instead of “turn ends” to emphasize that the ability resolves after the player discards to their maximum hand size. These conditions are now explicitly stated to be the same.
- Changed the definition of an action from any paid ability with [click] in its trigger cost to only paid abilities where the trigger cost begins with [click] ([section 5.2.1](#)).
- Removed the second sentence of [section 5.2.2](#), which was vague and redundant with [section 9.5.6](#).
- Clarified when a “take an action” trigger condition is met ([rule 5.2.2c](#)).
- Added quotation marks around the game text of each basic action (sections [5.2.7](#) and [5.2.8](#)).
- Updated the wording of the Corp’s “Trash 1 resource” basic action for improved clarity ([section 5.2.7](#)).
- Completely reworked and updated the timing structure of a run ([section 6](#)). See the Release Notes for details.
- Streamlined rules for **deflectors** and other effects that can move the Runner outside the normal progression of a run ([section 6.1.2](#) and [section 6.2.5](#)).
- Codified the use of “after” to refer to consecutive phases of a run ([rule 6.1.3e](#)).
- Added rules for fully breaking ice in an encounter ([section 6.5.7](#)).
- Completely reworked and updated the rules for accessing ([section 7](#)). Most importantly, the new keyword “breach” encapsulates the rules for accessing the set of cards associated with a particular server.
- Introduced mid-access abilities as a new kind of paid ability, and redefined the Runner’s opportunity to trash cards they access as an ability of this kind, called the “basic trash ability” ([section 7.1.5](#)).

- Added rules to handle the two cases (*Information Sifting* and *Counter Surveillance*) where an ability allows the Runner to access multiple cards in a different structure than a breach ([rule 7.1.9](#) and [rule 7.1.10](#)).
- Overhauled [section 8](#) to cover all changes to card locations and faceup/facedown status that have special rules, rather than focusing only on install, rez, and play.
- Consolidated the rules about replacement effects altering draw, score, steal, and trash effects into a unified rule covering all movements ([section 8.2.2](#)).
- Added rules for arranging and rearranging cards in [section 8.3](#).
- Introduced a new procedure for drawing cards that allows for cleaner handling of abilities like *Daily Business Show* and *Jinja City Grid* ([section 8.4](#)).
- Consolidated rules in [section 8.5.6](#) about trashing like cards when installing an agenda, asset, or upgrade.
- Generalized the rule for “Remote server only” to cover any type of server restriction ([rule 8.5.12](#)).
- Updated [step 8.6.6g](#) of playing an event/operation to only trash the card if it is still in the play area.
- Moved the rules for searching from [section 4](#) to [section 8.7](#). Moved the rules for swapping from [section 10](#) to [section 8.8](#).
- Subroutines are now their own ability type, covered in [section 9.8](#).
- Updated the rules about the source of an ability ([section 9.1.3](#)) and using an ability ([section 9.1.6](#)) for improved clarity.
- Updated the example for [rule 9.1.4](#) about an ability becoming independent of its source.
- Clarified the definition of a timing structure ([section 9.2.2](#)).
- Expanded the cases where an ability is active even though its source is inactive to handle abilities that look for their source to be trashed, subroutines on a piece of ice that is encountered without being installed, and persistent abilities ([section 9.1.8](#)).
- Added rules for mid-access ability windows, a new type of priority window for handling mid-access abilities ([section 9.2.10](#)).
- Added rules for ability flags ([section 9.3.6](#)). The ability flags are access, interface, [interrupt], and persistent.
- Expanded rules for when players can trigger paid abilities to cover mid-access and interface abilities ([section 9.5.2](#)).
- Clarified how *Mumbad Virtual Tour* and *Neutralize All Threats* interact with “using” mid-access abilities ([section 9.5.3](#)).

- Generalized [rule 9.5.5](#) regarding hosted objects and paid abilities that uninstall their source card, and updated its examples.
- Removed a rule just before [rule 9.6.10](#) that described how subroutines previously functioned as a special type of conditional ability.
- Expanded rules for delayed conditional abilities ([section 9.6.13](#)).
- Clarified how subroutines are gained and lost, the meaning of broken and unbroken subroutines, and how exactly *Grappling Hook* works ([section 9.8](#)).
- Expanded rules for lingering effects ([section 9.10](#)).
- Updated the list of ways a checkpoint can occur in the middle of resolving an instruction ([section 9.11.1](#)).
- Each step in a timing structure (but not in other defined sequences of steps) now counts as an instruction and follows the same rules as instructions from abilities ([section 9.11.2](#)). Removed explicit checkpoint steps from all the timing structures as they are now redundant. Also removed an extraneous checkpoint from the steps of playing an event/operation ([section 8.6.6](#)) and from the steps of resolving a trace attempt ([section 10.8.6](#)).
- Introduced the “persistent” keyword ([section 9.12.5](#)).
- Added a **console** check to [step 10.3.1d](#) of the checkpoint procedure. This limitation is now handled alongside uniqueness rather than grouped under the broader “card installed in an illegal location” step.
- Reworded [step 10.3.1e](#) of the checkpoint procedure to improve clarity.
- Clarified the presence of checkpoints after resolving an instruction ([rule 10.3.5](#)) and at the end of a timing structure ([rule 10.3.6](#)).
- Removed the list of card errata. Cards with updated rules text are now documented in the NISEI Card Text Updates Document, which can be found at <nisei.net/rules/card-text-updates>.
- Removed the appendix listing previous revisions. Older versions of this document are available [online](#).

Table of Contents

[Summary of Changes](#)

[Table of Contents](#)

1. [Game Concepts](#)

- 1.1. [General](#)
- 1.2. [The Golden Rules](#)
- 1.3. [Symbols](#)
- 1.4. [Deck Construction](#)
- 1.5. [Extra Cards](#)
- 1.6. [Starting the Game](#)
- 1.7. [Ending the Game](#)
- 1.8. [Cards](#)
- 1.9. [Counters and Tokens](#)
- 1.10. [Credits](#)
 - 1.10.5. [Recurring Credits](#)
- 1.11. [Clicks](#)
- 1.12. [Objects](#)
- 1.13. [Host, Hosted, and Hosting](#)
- 1.14. [Ownership and Control](#)
- 1.15. [Targets](#)
- 1.16. [Costs](#)
- 1.17. [Score, Scoring, and Stealing](#)
- 1.18. [Advancing Cards](#)
- 1.19. [Trashing](#)
- 1.20. [Memory](#)
- 1.21. [Card Visibility](#)

2. [Parts of a Card](#)

- 2.1. [Name](#)
- 2.2. [Unique Symbol](#)
- 2.3. [Play Cost, Install Cost, or Rez Cost](#)
- 2.4. [Advancement Requirement](#)
- 2.5. [Agenda Points](#)
- 2.6. [Trash Cost](#)
- 2.7. [Strength](#)
- 2.8. [Memory Cost](#)
- 2.9. [Base Link](#)
- 2.10. [Starting Memory Limit](#)
- 2.11. [Minimum Deck Size](#)
- 2.12. [Influence Limit](#)

2.13. [Faction Affiliation](#)

2.14. [Influence Cost](#)

2.15. [Card Type](#)

2.16. [Subtypes](#)

2.17. [Text Box](#)

3. [Card Types](#)

- 3.1. [Identities](#)
- 3.2. [Agendas](#)
- 3.3. [Assets](#)
- 3.4. [Ice](#)
- 3.5. [Operations](#)
- 3.6. [Upgrades](#)
 - 3.6.5. [Regions](#)
- 3.7. [Events](#)
- 3.8. [Hardware](#)
 - 3.8.5. [Consoles](#)
- 3.9. [Programs](#)
 - 3.9.5. [Icebreakers](#)
- 3.10. [Resources](#)

4. [Game Zones](#)

- 4.1. [General](#)
- 4.2. [Deck](#)
 - 4.2.7. [R&D](#)
 - 4.2.8. [Stack](#)
- 4.3. [Hand](#)
 - 4.3.8. [HQ](#)
 - 4.3.9. [Grip](#)
- 4.4. [Discard Pile](#)
 - 4.4.6. [Archives](#)
 - 4.4.7. [Heap](#)
- 4.5. [Score Area](#)
- 4.6. [Play Area](#)
 - 4.6.6. [Servers](#)
 - 4.6.7. [Central Servers](#)
 - 4.6.8. [Remote Servers](#)
 - 4.6.9. [Protecting a Server](#)
- 4.7. [Bank](#)
- 4.8. [Set Aside](#)
- 4.9. [Removed from the Game](#)

5. [Turns](#)
 - 5.1. [General](#)
 - 5.2. [Actions](#)
 - 5.2.7. [Corp Actions](#)
 - 5.2.8. [Runner Actions](#)
 - 5.3. [Draw Phase](#)
 - 5.4. [Action Phase](#)
 - 5.5. [Discard Phase](#)
 - 5.6. [Steps of the Corp's Turn](#)
 - 5.7. [Steps of the Runner's Turn](#)
6. [Runs](#)
 - 6.1. [General](#)
 - 6.2. [Position](#)
 - 6.3. [Initiation](#)
 - 6.4. [Approach Ice](#)
 - 6.5. [Encounter Ice](#)
 - 6.5.7. [Fully Break](#)
 - 6.5.8. [Bypass](#)
 - 6.5.9. [Forced Encounters](#)
 - 6.6. [Movement](#)
 - 6.7. [Success](#)
 - 6.8. [Run Ends](#)
 - 6.9. [Steps of a Run](#)
7. [Accessing Cards and Breaching Servers](#)
 - 7.1. [Accessing Cards](#)
 - 7.2. [Steps of Accessing a Card](#)
 - 7.3. [Breaching Servers](#)
 - 7.4. [Determining Candidates](#)
 - 7.5. [Steps of Breaching a Server](#)
8. [Card Manipulation](#)
 - 8.1. [Faceup and Facedown Status](#)
 - 8.2. [Card Movements](#)
 - 8.3. [Arranging and Rearranging Cards](#)
 - 8.4. [Drawing Cards](#)
 - 8.4.5. [Steps of Drawing N Cards](#)
 - 8.5. [Installing and Uninstalling Cards](#)
 - 8.5.13. [Steps of Installing a Card](#)
- 8.6. [Playing Events and Operations](#)
 - 8.6.6. [Steps of Playing an Event or Operation](#)
- 8.7. [Searching for Cards](#)
- 8.8. [Swapping Cards](#)
9. [Abilities](#)
 - 9.1. [General](#)
 - 9.2. [Timing and Priority](#)
 - 9.3. [Interpreting Card Text](#)
 - 9.4. [Static Abilities](#)
 - 9.5. [Paid Abilities](#)
 - 9.5.6. [Steps of Using a Paid Ability](#)
 - 9.6. [Conditional Abilities](#)
 - 9.6.14. [Steps of Triggering and Resolving a Conditional Ability](#)
 - 9.7. [Play Abilities](#)
 - 9.7.2. [Steps of Resolving a Play Ability](#)
 - 9.8. [Subroutines](#)
 - 9.8.10. [Steps of Resolving a Subroutine](#)
 - 9.9. [Interrupts and Replacement Effects](#)
 - 9.10. [Lingering Effects](#)
 - 9.11. [Identifying Instructions](#)
 - 9.12. [Other Rules and Terminology](#)
 - 9.12.1. [Simultaneous Effects](#)
 - 9.12.2. [Acting on Sets of Cards](#)
 - 9.12.3. ["Must"](#)
 - 9.12.4. ["Repeat this process"](#)
 - 9.12.5. ["Persistent"](#)
10. [Additional Rules](#)
 - 10.1. [General](#)
 - 10.2. [Information](#)
 - 10.2.2. [Hidden Information](#)
 - 10.2.3. [Open Information](#)
 - 10.3. [Checkpoints](#)
 - 10.4. [Damage](#)
 - 10.5. [Tags](#)
 - 10.6. [Bad Publicity](#)
 - 10.7. [Link](#)
 - 10.8. [Traces](#)

- 10.8.6. [Steps of Resolving a Trace Attempt](#)
- 10.9. [Load and Empty](#)
- 10.10. [Infinite Loops](#)
- 10.11. [Self-Referential Language](#)
- 11. [Appendix: Timing Structure Reference](#)
 - 11.1. [Overview](#)
 - 11.2. [Timing Structure of Turns](#)
 - 11.2.1. [Corp's Turn](#)
 - 11.2.2. [Runner's Turn](#)
 - 11.3. [Timing Structure of a Run](#)
 - 11.4. [Timing Structure of Breaching a Server](#)
 - 11.5. [Timing Structure of Accessing a Card](#)
- [Acknowledgements](#)

1. Game Concepts

1.1. General

These rules are compatible with cards from the game ANDROID: NETRUNNER by Fantasy Flight Games. ANDROID: NETRUNNER is a game about the cyber struggle between massive Corporations and subversive hackers known as Runners.

- 1.1.1. The game is played between two players. One player takes the role of the Corp (Corporation) and the other takes the role of the Runner. This rules document will frequently refer to a player interchangeably with their game role.
- 1.1.2. Each player needs a legal deck, an identity card for their role, and any extra cards used from outside their deck. They also need a supply of tokens as described in [section 1.9](#). The constraints that define the legality of a deck are defined in [section 1.4](#), and the cases where cards outside the deck and identity can be used are defined in [section 1.5](#).

1.2. The Golden Rules

- 1.2.1. If the text of a card directly contradicts these rules, the text of the card takes precedence.
- 1.2.2. If a rule or ability directs something to happen, but another effect states that it cannot happen, the “cannot” ability takes precedence.
 - a. If a “cannot” effect prohibits all of the effects of another ability, that ability cannot be triggered.
 - b. If a “cannot” effect prohibits only part of another ability, that ability can be triggered, but the prohibited steps of resolving that ability are not carried out.

Example: During a run, Lockdown’s subroutine resolves, preventing the Runner from drawing cards for the remainder of the turn. The Runner has a Diesel and a Process Automation in their grip. For the remainder of this turn, they cannot play Diesel as its entire ability is prohibited, but they can play Process Automation. Even though cards cannot be drawn through Process Automation, the Runner can play it to gain 2[c].

- 1.2.3. If an instruction includes the words “if able,” it can only be carried out fully or not at all. If any part of the instruction is not possible to carry out, the entire instruction is ignored.
- 1.2.4. If an instruction does not include the words “if able,” as much of that instruction as possible is carried out. Any parts of the instruction that are not possible to carry out are ignored.
- 1.2.5. A player can only take an action or use an ability if its effect has the potential to change the game state. This potential is assessed strictly by what the action or ability can be expected to accomplish, without regard to the consequences of paying any costs to

initiate that action or ability and without regard to any other abilities that may meet their conditions in the process of initiating or resolving that action or ability.

Example: The Corp has one unrezzed piece of ice installed and Liquidation in HQ. As Liquidation requires the Corp to have at least one rezzed card to trash, the Corp cannot play Liquidation as it cannot change the game state.

Example: The Corp has one rezzed piece of ice installed and Liquidation in HQ. Because the Corp has at least one rezzed card that could be trashed (which would change the game state), they can play Liquidation, spending a click and paying its play cost. The Corp then chooses any number of their rezzed cards to trash, which could be zero cards.

Example: The Runner is playing Armand “Geist” Walker and has Forger installed. The Runner can only trash Forger and trigger Geist’s ability when they are about to take a tag (which Forger could avoid) or while they have a tag (which Forger could remove). Using Forger at any other time has no potential to change the game state.

1.3. Symbols

- 1.3.1. Several non-English symbols appear on cards and in this rules document. This section serves as a basic guide to those symbols.
- 1.3.2. When this document is presented in a format without images, plaintext replacements are used. These replacements are listed along with the symbols themselves for reference.
- 1.3.3. The symbol [c] (plaintext: [c]) stands for “credit”. It always appears with a numeral, such as 1[c], which means “one credit,” or 3[c], which means “three credits.” See [section 1.10](#) for rules about credits.
- 1.3.4. The symbol [click] (plaintext: [click]) stands for “click”. Multiple clicks can be represented either by multiple symbols, such as [click][click], or by a numeral and symbol, such as 2[click], both meaning “two clicks.” See [section 1.11](#) for rules about clicks.
- 1.3.5. The symbol [recurring] (plaintext: [recurring]) stands for recurring credit. It always appears with a numeral, such as 1[recurring], which means “one recurring credit,” or 3[recurring], which means “three recurring credits.” See [section 1.10.5](#) for rules about recurring credits.
- 1.3.6. The symbol [link] (plaintext: [link]) stands for “link”. It is always used with a quantity, such as 1[link], which means “1 link.” See [section 10.7](#) for rules about link.
- 1.3.7. The symbol [MU] (plaintext: [MU]) stands for “memory unit”. It always appears with a quantity, such as 2[MU], which means “2 memory units.” See [section 1.20](#) for rules about memory.
- 1.3.8. The symbol [sub] (plaintext: [sub]) stands for “subroutine”. Each symbol marks a single subroutine on a piece of ice. See [rule 3.4.7](#) for information about subroutines.
- 1.3.9. The symbol [trash] (plaintext: [trash]) stands for “trash this card”. It is used as a self-referential cost in card text, such as “[trash]: Draw 2 cards,” which means “Trash this

card to draw 2 cards.” See [section 1.16](#) for rules about costs, and [section 1.19](#) for rules about trashing cards.

- 1.3.10. The symbol [interrupt] (plaintext: [interrupt]) stands for “interrupt timing”. It is used to designate abilities that resolve immediately before other abilities. See [section 9.9](#) for rules about interrupt abilities.

1.4. Deck Construction

- 1.4.1. Each player’s deck is associated with a single identity card that determines the faction, minimum deck size, and influence limit of that deck. The identity card may also stipulate other variances from the standard deckbuilding rules.
 - a. **The identities *The Catalyst* and *The Syndicate* from the *System Gateway Starter Pack* are intended for use only with the decks included in that pack. They are not legal for play under the full deck construction rules.**
- 1.4.2. Each deck must meet all requirements in this section to be legal for play.
- 1.4.3. The deck must contain at least as many cards as the minimum deck size indicated on the corresponding identity.
 - a. Identity cards, extra cards that begin the game outside the deck, and player aid cards are not part of the deck and are not counted towards the size of the deck.
 - b. There is no maximum deck size.
- 1.4.4. Decks cannot contain identity cards, player aid cards, cards from the wrong side (Corp cards in a Runner deck or vice-versa), or out-of-faction cards that lack influence costs.
- 1.4.5. Neutral cards and cards that belong to a faction other than the corresponding identity’s faction are all considered out-of-faction. The total influence cost of out-of-faction cards in the deck must not exceed the influence limit of that identity.
 - a. The total influence cost of out-of-faction cards is counted by copy and not by name.

Example: Including a single copy of Diesel in a non-Shaper deck adds 2 to the total influence in that deck, while including two copies of Diesel in a non-Shaper deck adds 4 to the total influence in that deck.
- 1.4.6. A Corp deck must contain agendas totalling a certain number of agenda points, as determined by the total number of cards in the deck.
 - a. A deck with 40 to 44 cards must contain 18 or 19 agenda points.
 - b. A deck with 45 to 49 cards must contain 20 or 21 agenda points.
 - c. A deck with 50 to 54 cards must contain 22 or 23 agenda points.
 - d. A deck with more than 54 cards must contain 22 or 23 agenda points, plus an additional 2 agenda points for every full 5 cards in the deck over 50.

Example: A 66 card deck requires 6 additional agenda points, since it includes 3 sets of 5 cards beyond 50. This gives a final requirement of either 28 or 29 agenda points.

- 1.4.7. The deck cannot contain more than 3 copies of any single card, by name. Some cards stipulate alternative copy limits in their card text.
- 1.4.8. Tournament play may impose other restrictions or requirements on the legal configurations of identities and decks.

1.5. Extra Cards

- 1.5.1. Some abilities allow for the use of additional cards from outside the deck.
- 1.5.2. One Corp identity, *Jinteki Biotech: Life Imagined*, has the ability, “Before taking your first turn, you may switch this card with any copy of Jinteki Biotech.” There are 3 versions of this identity, which have different abilities on their reverse side.
 - a. A player playing Jinteki Biotech as their identity may bring any number of copies of that identity along with their deck. After completing game setup, that player may choose any copy they own to be their active identity for the duration of the game. All other copies are placed outside the game.
- 1.5.3. One Runner identity, *Adam: Compulsive Hacker*, has the ability, “You start the game with 3 different **directive** cards installed (these cards are not considered part of your deck).”
 - a. A player playing Adam as their identity must bring at least 3 differently named cards with the **directive** subtype along with their deck. The player may bring any number of **directive** cards over the required 3.
 - b. After players reveal their identities, the player playing Adam selects exactly 3 of their provided **directive** cards. Those cards begin the game installed in the play area. All other **directives** the player brought this way remain outside the game.
 - c. A player playing Adam can also include **directive** cards in their deck. The cards chosen to be installed at the beginning of the game do not impact the deck’s influence requirements or the maximum allowed number of copies of those cards.
 - d. Once the game has begun, the cards installed this way are treated exactly as any other installed cards. Abilities that move these cards to other zones function normally, including shuffling them into the stack.
- 1.5.4. Two cards, *Rebirth* and *DJ Fenris*, make use of identity cards other than the one selected by the player during deck construction.
 - a. A player may bring any number of additional Runner identity cards along with their deck. These cards are kept in a pile outside the game. The Runner may look at these cards at any time.

- b. When an ability refers to an identity other than the Runner's current identity, it refers to the cards provided this way. If an identity card leaves the play area, it must be returned to the pile outside the game.
- c. In a tournament, all identities brought to the game this way must be legal for players to use as their actual identity in that tournament.
- d. One Runner identity, *Hoshiko Shiro*, is double-sided (see [section 3.1](#)). If the Runner switches their identity with Hoshiko, it enters play with the front side faceup. If the Runner switches Hoshiko with another identity, the new identity enters play faceup regardless of Hoshiko's current status.

1.6. Starting the Game

- 1.6.1. The players decide who will play as the Corp and who will play as the Runner. Each player places an appropriate identity card for the side they are to play faceup in the play area, then supplies a corresponding deck, placing it facedown in the play area, and any appropriate extra cards.
 - a. Some identity cards have abilities that affect setup. Even though cards are not active until the game begins, these identities still alter setup at the appropriate step as indicated by their text.
 - b. Some identities are double-sided (see [section 3.1](#)). If a player is playing a double-sided identity, it begins the game with the front side faceup.
- 1.6.2. If a player's identity has an ability that affects setup or the start of the game, and that ability does not directly correspond to a setup step outlined in this section of the rules, that player makes any necessary decisions and changes for their identity's special setup or start of game abilities at this time. If both players have setup or start of game abilities at this time, the Corp resolves theirs first.
- 1.6.3. The players create the bank by gathering all types of tokens described in [section 1.9](#).
- 1.6.4. Each player takes five credits from the bank, placing the credits in their credit pool.
- 1.6.5. Each player shuffles their deck.
- 1.6.6. Each player draws 5 cards from the top of their deck to form their starting hand.
 - a. After drawing starting hands, the Corp may choose to take a mulligan; then, the Runner may choose to take a mulligan. To take a mulligan, the player shuffles their starting hand back into their deck, then draws a new starting hand. They must keep the second hand as their starting hand.
- 1.6.7. The game **begins** and the Corp takes their first turn.
 - a. If the Corp's identity has an ability with instructions for "before taking your first turn," the Corp resolves that ability immediately before taking their first turn, and thus before the game starts.

1.7. Ending the Game

- 1.7.1. The game ends when a player meets one of their win conditions.
 - a. If both players would simultaneously satisfy their win conditions, the game ends in a draw.
- 1.7.2. Each player has two possible win conditions available.
 - a. Either player can win by collecting agenda points in their score area (usually done by scoring or stealing agendas; see [section 1.17](#)). A player with a score of 7 or more wins the game.
 - b. The Corp wins if the Runner is flatlined. The Runner is flatlined immediately if they suffer more damage than they have cards in their grip. The Runner is also flatlined if, at the beginning of their discard phase, their maximum hand size is less than 0. See [section 10.4](#) for more information about damage.
 - c. The Runner wins if the Corp is required to draw a card from R&D but cannot because R&D is empty.

1.8. Cards

- 1.8.1. There are six types of Corp cards: agendas, assets, ice, identities, operations, and upgrades. All cards except the identity card are shuffled into the Corp's deck at the beginning of the game.
- 1.8.2. There are five types of Runner cards: events, hardware, identities, programs, and resources. All cards except the identity card are shuffled into the Runner's deck at the beginning of the game.
- 1.8.3. Cards that are ACTIVE are able to affect the game through their abilities.
 - a. Runner cards that are installed and faceup in the play area, Corp cards that are installed and rezzed, events and operations in the play area, agendas in the Corp's score area, and both players' identities are active.
 - b. Unless otherwise stated, Runner cards **become active as part of the process of playing or installing them**.
 - c. Unless otherwise stated, Corp cards other than operations are installed unrezzed, and thus inactive, into the play area. Assets, ice, and upgrades **become active when they are rezzed**; agendas **become active when they are scored**. Operations **become active as part of the process of playing them**.
 - d. **Identity cards become active when the game begins.**
- 1.8.4. Cards that are INACTIVE are unable to affect the game or have most of their abilities used.

- a. Cards are inactive in R&D, HQ, Archives, the heap, the grip, the stack, the Runner's score area, while set aside, and while removed from the game. Cards installed facedown are also inactive.
 - b. Inactive cards in most zones retain their printed characteristics (name, card type, faction, cost, subtypes, influence, etc). Runner cards installed facedown have no characteristics.
- 1.8.5. For a player to DRAW 1 or more cards is to take that many cards from the top of their deck and put them into their hand. **Section 8.4 provides more details about drawing cards.**
- 1.8.6. Most abilities are active if and only if the card they appear on is active. **Section 9.1.8** details the cases where abilities on inactive cards are still active.
- 1.8.7. Abilities can convert cards from one type into another type, or even from a card into a counter. **Rules 10.1.3** and **10.1.4** explain the details of card conversion.

1.9. Counters and Tokens

- 1.9.1. COUNTERS and TOKENS are game pieces (or equivalent) that track various resources, effects, and statuses of players and their cards.
- a. The terms “counter” and “token” are interchangeable.
- 1.9.2. The BANK is the supply of **counters** not yet in play. **Counters** in the bank are available to both players to take and use as dictated by the game rules and card abilities. Players do not control **counters** in the bank and cannot spend them. The bank is an unlimited supply; running out of game pieces to track a type of **counter** does not prohibit a player from gaining **counters** of that type or placing them on cards.
- a. **If an instruction directs players to place counters on a card without designating where those counters come from, take those counters from the bank.**
 - b. **A condition or restriction that looks for counters being placed on a card can be met by either counters from the bank or counters moved from another location.**
- 1.9.3. Some abilities can convert a card into a counter. A counter put into play in this way is tracked with the card acting as a game piece. See **rule 10.1.4**.
- 1.9.4. Some abilities can cause a player or card to have or be “considered to have” 1 or more **counters** or additional **counters** without placing or giving those **counters**. A **counter** considered to exist in this way conveys all the same information and effects as a **counter** of the specified type would, except that it cannot be moved or removed from the card or player it belongs to, nor can it be spent to pay a cost.

- 1.9.5. There are ten types of **counters**: credit **counters**, click **counters**, tag **counters**, bad publicity **counters**, brain damage **counters**, advancement **counters**, virus counters, power counters, agenda counters, and condition counters.
- a. CREDIT **COUNTERS** are used to track the number of credits each player has in their credit pool; they can also be placed on cards. Rules for credits, the credit pool, and other related concepts are in [section 1.10](#).
 - b. CLICK **COUNTERS** are a gameplay aid used to track the clicks the active player has spent or has left to spend during their turn. Rules for clicks are in [section 1.11](#).
 - c. TAG **COUNTERS** are used to represent tags on the Runner. Rules for tags are given in [section 10.5](#).
 - d. BAD PUBLICITY **COUNTERS** represent bad publicity the Corp has earned. Rules for bad publicity are in [section 10.6](#).
 - e. BRAIN DAMAGE **COUNTERS** are a gameplay aid used to track brain damage the Runner has suffered. Each point of brain damage the Runner suffers forces them to take a brain damage token. See [section 10.4](#).
 - f. ADVANCEMENT **COUNTERS** are a **counter** used primarily on installed agendas to track the Corp's progress toward being able to score them.
 - g. VIRUS COUNTERS are a generic **counter**, used primarily by **virus** programs, that can be removed by the Corp purging them. See [rule 10.1.2](#).
 - h. POWER COUNTERS are a generic **counter** used by a variety of cards. They have no special rules.
 - i. AGENDA COUNTERS are a generic **counter** used primarily on scored agendas. They have no special rules.
 - j. CONDITION COUNTERS are **counters** that have rules text. Their abilities are active as long as they are hosted on a card.

1.10. Credits

Each of the Runner's credits represents enough money to upgrade some basic parts for their console, have a meal at a decent restaurant, or buy a ticket and some concessions for a night at the sensies.

Each of the Corp's credits represents enough money to manufacture a run of computer parts, buy out a decent restaurant, or film a low-budget sensie.

- 1.10.1. A CREDIT ([c]) is the basic unit of currency. Players spend their credits to pay for various costs, card abilities, traces, etc. Credit **counters** most commonly represent 1[c] each, but can represent larger denominations if clearly marked.

1.10.2. Each player has a CREDIT POOL where they keep a supply of credit **counters** matching the credits they have available to spend. The number of credits in a player's credit pool is open information.

1.10.3. Credits enter and leave a player's credit pool as that player gains, **loses**, or **spends** credits.

- a. A player GAINS credits whenever credits enter their credit pool from any location.
- b. If a player is **required** to LOSE credits, that player is forced to move the specified number of credits from their credit pool to the bank. Players cannot lose credits from cards.

Example: If a subroutine on DNA Tracker resolves, the Runner must lose 2[c] from their credit pool, or 1[c] if that is all they have. If the Runner's credit pool is empty, the effect does nothing.

- c. **To SPEND or PAY credits, a player puts** the specified number of credits back into the bank. **By default, the credits must come from the player's credit pool. Abilities can also allow players to spend credits hosted on their cards under certain conditions. If any such abilities apply, the player chooses how to divide the credits they are spending from among the allowed locations.** The terms "spend" and "pay" are synonymous.

Example: The Runner may use the credit from Cyberfeeder to pay for Atman's first ability.

Example: The Runner may use credits from Ghost Runner when secretly spending credits to resolve a Psi ability (such as The Future Perfect).

Example: The Runner must use credits from Ghost Runner if this is the only way for them to pay 3[c] when resolving the "when encountered" ability on Tollbooth.

1.10.4. Abilities can place credits on cards, as discussed in [section 1.13](#). Credits on a player's card are not in that player's credit pool; the player can only interact with those credits as instructed by card abilities.

- a. If a player is instructed to TAKE credits from a card, they remove that many credits from that card and gain those credits.
- b. If an ability specifies how a player is allowed to spend credits from a card, they can be spent from that card as if they were in the player's credit pool for the specified actions or abilities.
- c. If an ability specifies a time period during which a player is allowed to spend credits from a card, they can be spent from that card as if they were in the player's credit pool for any purpose (other than losing credits) as long as the specified time period is active or the specified duration has not expired.

- d. Spending credits from a card is considered “using” the card that allowed those credits to be spent. See [section 9.1.6](#).

1.10.5. Recurring Credits

- a. RECURRING CREDITS ([recurring]) place credits on a card repeatedly. The text “N[recurring]” means **“When this card becomes active, place N credits on it. Before abilities meet their trigger conditions for your turn beginning, if there are fewer than N credits on this card, place credits on it until there are N credits on it.”**
- b. **Recurring credits are first placed on a card as soon as it becomes active. This occurs at [step 8.5.13e](#) of installing the card if it is installed active, [step 8.6.6c](#) of playing an event or operation, or as soon as the card is turned faceup or scored, as applicable.**
- c. Recurring credits are refilled during [step 5.6.1c](#) of the Corp’s turn and [step 5.7.1c](#) of the Runner’s turn, before other abilities that apply at the start of the turn resolve.
- d. Recurring credits do not accumulate. They are refilled only up to the indicated number.

Example: The Runner installs Spinal Modem, which has 2[recurring]. 2 credits are placed on Spinal Modem now. The Runner spends 1 of those credits later in their turn. At the beginning of the Runner’s next turn, the credits on Spinal Modem should be replenished up to 2[c], so 1 more credit is placed on it.

1.11. Clicks

Working a job, making connections, and especially jacking in—everything you do takes time, and it always goes by faster than you think. A click represents an abstract amount of time spent on a particular activity, either several hours all at once or scattered across the day.

- 1.11.1. A CLICK ([click]) is the basic unit of activity. Players spend their clicks to perform actions and trigger abilities. Each click **counter** represents 1 click.
- 1.11.2. As the first step of a player’s turn, they gain an allotted number of clicks to spend during the action phase of that turn. See [section 5](#) for details about the procedures of player turns.
 - a. The Corp receives 3[click] on each of their turns.
 - b. The Runner receives 4[click] on each of their turns.

1.11.3. **Players can gain, lose, or spend clicks.**

- a. **A player GAINS clicks whenever the number of clicks they have is increased.**

- b. If a player **LOSES** or **SPENDS** clicks, the number of clicks they have is reduced by that amount. The terms “lose” and “spend” are not synonymous for the purposes of meeting conditions and restrictions of card abilities.
 - c. Paid abilities that begin with [click] are actions and have special timing rules (see [section 5.2](#)). Aside from taking actions, there are no special restrictions to how or when a player can spend their clicks during their turn.
- 1.11.4. Some cards have the subtype **priority** and the text, “Play only as your first [click].” A player can only play a **priority** card using the basic action to play an event or operation, and only if they have not spent any other clicks that turn. Losing clicks does not affect a player’s ability to play a **priority** card.

1.12. Objects

- 1.12.1. An **OBJECT** is an occurrence of a card or counter existing in a particular zone. When a card or counter moves from one zone to another, it becomes a new object.
- 1.12.2. Restrictions and effects on a card or counter only apply for as long as that card or counter remains the same object. Cards and counters have no memory of the objects they were in the past.

Example: The Corp uses the first ability on Vaporframe Fabricator, which specifies that it can only be used once per turn, and trashes Vaporframe Fabricator during the installation process. Then the Corp plays Restore, reinstalling the same copy of Vaporframe Fabricator from Archives. Even though it is the same physical card, the reinstalled Vaporframe Fabricator is a new object, and its “once per turn” ability is available to be used this turn.

Example: The Runner accesses Ganked! from the root of a server they are breaching. The Corp uses Ganked!’s ability to trash it and force the Runner to encounter Drafter. While resolving Drafter’s subroutines during that encounter, the Corp reinstalls the same copy of Ganked! from Archives into the same root. Since the Ganked! card left and reentered the zone it was accessed from, it is now a new object, and the Runner must access that object before completing the breach. Thus, if the Runner does not interfere (such as by breaking Drafter’s subroutines), the Corp could repeat this sequence of events several times.

- a. An ability that moves cards can identify the new objects that those cards become and continue to act on them.

Example: The Corp plays Priority Construction, installing a piece of ice from HQ. That ice is now a new object in the play area, but Priority Construction’s second instruction can still find it and place advancement counters on it.

- 1.12.3. A card that moves from its location to a hidden or secret location becomes a new object, even if the new location is in the same zone.

Example: The Corp scores Accelerated Beta Test, looks at the top 3 cards of R&D, and installs and rezzes a piece of ice from among them. As a chain reaction, the Corp then uses The Foundry: Refining the Process to search R&D, with the result that the other cards currently being looked at by Accelerated Beta Test are shuffled along with the rest of R&D. Since the cards were moved to an unknown location in R&D, Accelerated Beta Test can no longer act on them, so those cards can no longer be looked at and will not be installed or trashed.

Example: The top card of R&D in the current game state is an object. So is the 2nd card from the top of R&D, the 3rd card, and so on. If the Corp plays Precognition and rearranges the top 5 cards of R&D, the Runner will not know which cards correspond to which objects in the previous order, so the cards become new objects under their new order.

- 1.12.4. A card or counter that moves to a known location in the same zone is still the same object.

Example: The Runner has chosen a piece of ice with Femme Fatale, and the Corp uses Thimblorig's ability to move that ice to another location. But the ice has not left the play area, so it is still the same object and the Runner can still use Femme Fatale's ability to bypass it.

- 1.12.5. A card that is turned faceup or facedown is still the same object.

Example: The Corp uses Vaporframe Fabricator, installing a piece of ice at no cost, and then plays Divert Power to derez Vaporframe Fabricator and rez their newly-installed ice at a discount. The Corp can rez Vaporframe Fabricator again, but since it is still the same object, its "once per turn" ability cannot be used again this turn.

- 1.12.6. When a card or counter becomes a new object, its previous object ceases to exist, but can still be remembered by rules or abilities that refer to past game states.

Example: The Runner plays Bravado. During the run, a piece of ice that the Runner has already passed is trashed. When the run ends, Bravado's ability must determine how many pieces of ice the Runner passed during that run. It does this by reviewing the game history and counting each distinct ice object that the Runner passed. Even though the object corresponding to the trashed ice no longer exists in the present game state, Bravado can still count it.

Example: The Runner trashes the top card of R&D while accessing it. The object corresponding to that card in its location on top of R&D ceases to exist, but still counts against the number of cards the Runner can access from R&D during this breach.

1.13. Host, Hosted, and Hosting

- 1.13.1. Whenever an **object** is "placed" or "loaded" onto a card, a host relationship is created between those two objects. The object placed on top of a card is HOSTED on that card. The card onto which the other object is placed is the HOST.

- 1.13.2. The state of being hosted is distinct, but not exclusive from, the state of being installed. If an ability instructs a player to host a card from an inactive state without reference to installing it, the hosted card remains inactive.
- 1.13.3. **An object** can only be hosted on a single card at any given time.
- 1.13.4. A card can host any number of other objects, except where explicitly restricted by a card ability.
- 1.13.5. Any card can act as a host, but a host relationship can only be created or permitted by a card ability.
- a. If a card has an ability describing the types and/or numbers of cards it can host, but it does not have an ability that directly hosts cards onto itself, then the card is permitted as an eligible installation destination for the types of cards listed, up to the number specified. A host card is chosen as an installation destination during [step 8.5.13b](#) of the installation process.

*Example: Off-Campus Apartment has the ability, “Off-Campus Apartment can host any number of **connections**.” This means that whenever the Runner installs a **connection** card, they can choose to install that card into the play area hosted on Off-Campus Apartment or as normal directly into the play area.*

- b. If a card has an ability describing cards it can host, but also one or more abilities that can create host relationships onto itself, then it only allows cards to become hosted on it through those abilities. A card cannot be installed directly onto the card through an unrelated install effect.

Example: Glenn Station reads, “Glenn Station can host a single card”, but also has a paid ability that hosts a card on itself. Therefore, Glenn Station only lets the Corp host cards through that paid ability, not through an install action.

- c. If a card has an ability stipulating that it can only be installed hosted onto another card, then during the installation process a player that wishes to install that card must choose a valid destination matching the description provided. If no such destination exists before the installation process begins, the card cannot be installed.

*Example: Egret states, “**Install only on a rezzed piece of ice.**” The Runner can only install Egret if there is an installed, rezzed piece of ice available before the installation process begins; otherwise, it is illegal for the Runner to install Egret.*

- d. Some operations and events convert themselves into condition counters and install themselves hosted onto other cards. An operation or event can never be installed through an install effect; it can only become an installed counter through its conversion ability. See [rule 10.1.4](#).

- 1.13.6. Unless otherwise noted, a card that is hosted as it is installed is hosted onto the destination card in the same faceup or facedown status it would normally be installed in.

- 1.13.7. Once a card is installed, hosted or not, a player cannot change that card's hosted status unless a card ability explicitly directs it to be hosted onto a new card.
- 1.13.8. Host relationships are not transitive. If card A is hosted on card B, and card B is hosted on card C, card A is not considered to be hosted on card C.

Example: If the Runner installs a Leprechaun hosted on a Dhegdheer, programs hosted on Leprechaun do not have their install costs reduced because they are hosted on Leprechaun and not on Dhegdheer.

- 1.13.9. If an ability on a card refers to a "hosted" card or counter, that ability only references **objects** hosted on that card.
- 1.13.10. Hosted **objects** can be removed or spent from their host without affecting the host.
- 1.13.11. If a host card is uninstalled, all **objects** hosted on that card (and all **objects** hosted on those **objects**, and so on) are trashed during the next checkpoint. This cannot be prevented. See [section 10.3](#) for details on checkpoints.

1.14. Ownership and Control

- 1.14.1. The OWNER of a card is the player who provided that card at the start of the game as their identity, part of their deck, or an extra card along with their deck.
 - a. A card that is converted into a counter retains its owner.
 - b. No player owns counters that are not converted cards.
- 1.14.2. The CONTROLLER OF **AN OBJECT** is the player responsible for that **object**.
 - a. The controller of a card in the play area is the player who installed or placed it there.
 - b. The Corp controls each agenda in the Corp's score area. The Runner controls each agenda in the Runner's score area.
 - c. Cards in other zones are controlled by their owner.
 - d. Each player controls the credits in their credit pool.
 - e. The Corp controls each bad publicity token. The Runner controls each tag token. However, see [rule 10.5.5](#) and [rule 10.6.3](#) regarding paying costs that involve tags and bad publicity, respectively.
 - f. The controller of a counter hosted on a card is the player who controls the host card.
 - g. Click tokens and brain damage tokens are gameplay aids only and have no controller.

- 1.14.3. A player can only pay costs using **objects** they control.
- 1.14.4. Unless otherwise stated, a player can only use abilities on **objects** they control.
- 1.14.5. The CONTROLLER OF AN ABILITY is the player who triggered or is allowed to trigger that ability. By default, this is the player who controls the ability's source. See also [section 9.1.3](#).
 - a. Some abilities state that they can only be used by a specific player. The specified player controls each such ability, even if they do not control its source.
- 1.14.6. Unless otherwise noted, the controller of an ability carries out its effects and makes any choices required. If a player is specified to perform all or part of an effect, the specified player carries out that part of the effect and makes any choices required instead.

*Example: Rototurret's first subroutine reads, "Trash 1 **installed** program." Since Rototurret is a Corp card, the Corp chooses which installed program to trash. Conversely, Bulwark's first subroutine reads, "The Runner trashes 1 **installed** program." Since the Runner is specified to be carrying out this effect, they choose which of their installed programs to trash.*

- a. Some trigger conditions care about effects performed by a particular player. These conditions are only met when that player is the one to carry out the relevant effect.

Example: The Corp has a rezzed Hostile Infrastructure. Apocalypse does not specify who trashes the cards, so the Runner is responsible for the trashing and Hostile Infrastructure meets its trigger condition.

*Example: Alice Merchant states that "the Corp must trash 1 card from HQ", so the Corp carries out this effect even though Alice Merchant is a Runner card, and therefore Hostile Infrastructure does not **meet its trigger condition**.*

1.15. Targets

- 1.15.1. Many abilities require players to choose 1 or more **objects or subroutines** to affect. Each **object and subroutine** chosen for an ability is a TARGET of that ability.

Example: Top Hat reads in part, "you may choose 1 of the top 5 cards of R&D and access it." The target of this instruction is the card in R&D the Runner chooses.

Example: One of the subroutines on Colossus reads in part, "Trash 1 installed program and 1 installed resource." The targets of this subroutine are the two cards that will be trashed.

*Example: Trick of Light reads, "**Choose 1 installed card you can advance. Move up to 2 advancement counters from 1 other card to the chosen card.**" The targets of this operation are the advancement **counters** to be moved and the destination card. If 2 tokens are chosen, they must be hosted on the same card.*

Example: The Runner is encountering a barrier and uses Cleaver to break subroutines. The targets of Cleaver's interface ability are the 1 or 2 subroutines that it will break.

- a. Cards found during a search are not considered targets of the instruction that performs the search. Players do not need to announce what card(s) they expect to find before the search actually takes place. See [section 8.7](#) for rules about searching.
 - b. Only **objects and subroutines** are announced as targets. If an instruction directs a player to choose (or "name") a number, a card type, a subtype, a card name, a server, or one of a specified set of effects, **that choice is** not made until the instruction resolves.
- 1.15.2. Immediately before an instruction becomes imminent, the player resolving that instruction must first choose the targets required by that instruction. For each time the instruction requires a player to choose **1 or more objects and/or subroutines**, that player ANNOUNCES appropriate targets. Once players are finished announcing targets for an instruction, the instruction becomes imminent, and players have the opportunity to modify the effects of that instruction with interrupt abilities before it resolves. See [section 9.9](#).
- a. When a player announces a card as a target, they indicate clearly which card is being chosen without changing its facedown/faceup status or location.
 - b. The player resolving the instruction can only choose **targets that are valid for that instruction**. A target is valid for an instruction if it meets any specified requirements of that instruction or can otherwise be acted upon by that instruction.
 - c. Unless an instruction explicitly specifies the zone from which **an object must be selected as a target**, only **counters in the play area and installed cards** are valid targets for that instruction.

Example: The Corp resolves a subroutine that says “[sub] The Runner trashes 1 program.” The Runner must trash 1 of their installed programs and may not trash a program from their grip or stack.

- d. Some effects allow a player to choose multiple cards as targets for a single announcement. [Section 9.12.2](#) contains rules about resolving effects involving sets of cards.
- e. A player may announce the targets for an instruction in any order, but each **object or subroutine** can only be chosen as a target once for each announcement. If there are not enough valid targets available, then that player chooses as many distinct targets as possible and the remaining targets are not announced.

*Example: The Runner accesses an Aggressive Secretary with three advancement **counters** on it. The Runner only has two installed programs, so the Corp chooses both of those programs for Aggressive Secretary. After the programs have been chosen, they are both trashed simultaneously.*

- f. Targets cannot be chosen for an instruction at any other time. Even if the game state changes due to interrupt abilities while the instruction is imminent, unannounced targets remain unchosen.
- 1.15.3. At the time an instruction resolves, if any of its targets either have become invalid or were not announced, as much of that ability resolves as possible without acting on the unannounced or invalid targets, following rules [1.2.3](#) and [1.2.4](#) of the Golden Rules. A target can be invalid because the chosen **object or subroutine no longer exists** or because the target no longer meets other requirements specified by the targeting ability.
- 1.15.4. After the **resolution of an** instruction that **chose a particular target**, subsequent instructions of the same ability can continue to act on that target without needing to select it again, even if the ability moves the target or changes its properties. **See also [section 1.12.2](#).**

Example: Howler's ability targets a card in HQ. Its first instruction installs that card, and its second instruction creates a delayed conditional ability that refers to that card in the play area. The latter ability can find and act on the card as long as it is still installed.

1.16. Costs

- 1.16.1. In order to initiate an action or ability or use an optional ability, a player may be required to pay a cost. A cost can take the form of any card, counter, or other item a player must spend; an effect a player must resolve; or other type of requirement a player must meet. If a player cannot pay the full cost of an action or ability all at once with cards and counters they control, they cannot use the effect associated with that cost.

- a. The act of paying a cost cannot be modified or cancelled by optional prevent/avoid abilities.

Example: The Runner can trash Clone Chip to trigger its paid ability, but cannot use LLDS Energy Regulator to prevent Clone Chip from being trashed this way.

- b. If a static ability or a mandatory conditional ability interrupt would prevent the steps of paying a cost from being carried out if they were performed as an effect, that cost cannot be paid.

Example: The Runner cannot pay the additional cost to steal Obokata Protocol while Guru Davinder is installed, because the Runner could not currently take 4 damage if instructed to: Guru Davinder's first ability would prevent the damage.

Example: The Runner is playing as Jesminder Sareen and encounters Funhouse, without any other tag-related effects resolving earlier in the run. Since Jesminder's ability would prevent taking 1 tag, the Runner cannot take 1 tag to pay the nested cost in Funhouse's ability. Funhouse will end the run.

- c. If a cost is subject to effects modifying its value, determine its final value from its default value by first applying each effect that increases the cost, then applying

each effect that lowers the cost. Finally, if the value determined by this process is less than 0, set the value to 0.

- d. A cost of 0 can still be paid by a player, even though no items are paid. **Costs of 0 are not paid automatically. A player pays a cost of 0 by explicitly announcing they are paying it.**

*Example: **The Runner** accesses a **Beanstalk Royalties** from R&D **and announces that they are using the ability of Freedom Khumalo: Crypto-Anarchist** to trash the **Beanstalk Royalties**. Even though the Runner does not remove any virus counters from any cards, they have still paid the cost.*

- 1.16.2. After a player pays a cost, a checkpoint occurs. See [rule 10.3](#).
- 1.16.3. Some cards have a printed play or install cost of X. The value of X is determined when the card is played or installed, after applying any modifiers which affect the play or install cost of that card.
 - a. If the play or install cost of a card with a play or install cost of X needs to be referenced at any other time, X is considered to be 0.
- 1.16.4. There are six main types of costs: install costs (found mainly on hardware, programs, and resources, and incurred when installing ice), play costs (found on operations and events), rez costs (found on assets, upgrades, and ice), paid ability trigger costs, additional costs, and nested costs.
 - a. Install costs, rez costs, and play costs are all costs that are inherent properties of cards. These costs are automatically applied to any ability that performs the corresponding **effects** (installing, rezzing, or playing) unless that ability stipulates otherwise. See also [section 2.3](#) (for more about these costs as they appear on cards) and [section 8.5.11](#), [rule 8.1.2d](#) and [section 8.6.2](#) (for more about paying costs to install, rez, and play cards).
 - b. When a player takes an action that plays or installs one or more cards and has no other effects, the play cost or install cost of each of those cards is considered to have been spent to take that action, even though other steps take place between initiating the action and paying that cost.

*Example: The Corp has **Jeeves Model Bioroids** rezzed, and uses the basic action "[Click]: Play an operation." to play **Blue Level Clearance**. Both the click spent for the cost of the basic action and the click spent for **Blue Level Clearance's** additional cost count as clicks spent to take the "[Click]: Play an operation." action.*

- 1.16.5. The INSTALL COST of a card must be paid to install that card.
 - a. The install cost of a program, resource, or piece of hardware is listed on the card itself.
 - b. The install cost of a piece of ice is a number of credits equal to the number of pieces of ice already protecting the server that ice will be installed protecting.

- c. There is no cost to install an asset, upgrade, or agenda.
- 1.16.6. The PLAY COST of an operation or event must be paid to play that card. It is listed on the card itself.
- 1.16.7. The REZ COST of an asset, upgrade, or piece of ice must be paid in order to rez that card. It is listed on the card itself.
- 1.16.8. The TRIGGER COST of each paid ability is the first part of that ability's text. It is followed by a colon (:) and then the ability's effect. See [section 9.5](#) for more details on paid abilities.
- 1.16.9. An ADDITIONAL COST adds something to the regular cost of initiating a particular game effect. A player must pay all additional costs along with any regular costs in order to initiate an effect.
 - a. If a player would be forced to carry out a game effect, but doing so has an additional cost, that player may decline to pay the additional cost, even if they are able to pay it, thus preventing that effect from occurring.

*Example: The Runner accesses Obokata Protocol. Normally the Runner **must** steal agendas that they access, but because Obokata Protocol has an additional cost to steal, the Runner can decline to suffer the net damage and not steal the agenda.*

- b. A player must pay all additional costs simultaneously with the cost that is being added to, even if multiple cards or abilities are adding separate additional costs. A player cannot pay the original cost or any of the additional costs individually. If they cannot pay for all of the costs at once, then they do not pay any of the costs and the corresponding effects cannot occur.

Example: The Runner accesses Obokata Protocol while Ben Musashi and Predictive Algorithm are both active. The Runner must be able to pay 2 credits and suffer 6 net damage all at once in order to steal the Obokata Protocol, even though each of the three costs are from different card abilities. After all costs have been paid, abilities that meet their trigger conditions from the paying of any of those costs, such as I've Had Worse or Order of Sol, can then resolve as applicable.

- c. If a player is directed to carry out a game effect and ignore a particular cost, they must still pay any additional costs that apply. If a player is directed to carry out a game effect "ignoring all costs", they do not pay any additional costs that would otherwise apply.**
- 1.16.10. A NESTED COST is a cost appearing within an ability's instructions that must be paid while the ability is resolving in order for some or all of the rest of the effects of that ability to resolve.
 - a. Nested costs are usually written in the format "[player] may [cost] to [instructions]" or "you may [cost]. If you do, [instructions]." If the indicated player pays the

indicated cost, the indicated instructions are resolved. Otherwise, that part of the ability is not resolved.

- b. Nested costs can also be written in the format “[instructions] unless [player] [cost].” If the indicated player pays the indicated cost, the indicated instructions are not resolved. Otherwise, that instruction is resolved.

Example: A subroutine reading “End the run unless the Runner pays 1[c].” contains a nested cost. If the Runner chooses to pay 1[c], the subroutine will not end the run.

1.17. Score, Scoring, and Stealing

- 1.17.1. The sum of all agenda points on agendas in a player’s score area is that player’s SCORE.
- 1.17.2. If at any time a player’s score becomes greater than or equal to 7, they win the game at the next checkpoint. See [section 1.7](#) regarding winning the game, and [section 10.3](#) for details about checkpoints.
- 1.17.3. Players add agendas to their score areas by SCORING or STEALING them. The Corp **can score** an agenda to their score area as an option during certain paid ability windows on their turn. The Runner **can steal** an agenda to their score area **while** accessing it.
 - a. The ADVANCEMENT REQUIREMENT of an agenda restricts when the Corp can score it. The Corp can only score an agenda that has advancement **counters** on it greater than or equal to its advancement requirement. See [section 1.18](#).
 - b. The Corp is not required to score an agenda immediately upon satisfying its advancement requirement.
 - c. Scoring an agenda does not cost [click] and is not an action.
 - d. The Runner cannot decline to steal an agenda they access, unless there is an additional cost to steal that agenda and the Runner cannot or does not wish to pay it. [Section 7](#) details accessing, and [section 1.16.9](#) details additional costs.
 - e. If an effect directly adds an agenda to a score area or otherwise moves an agenda from any zone to a score area, that agenda is not considered scored or stolen.
 - f. If an effect causes another card to be added to a score area “as an agenda”, the newly converted agenda is not considered scored or stolen. See [rule 10.1.3](#).
- 1.17.4. Agendas are always added to the score area faceup, regardless of their previous faceup/facedown status.
- 1.17.5. If an installed agenda is scored or stolen, it becomes uninstalled. Any advancement **counters** on the scored or stolen agenda are returned to the bank. See [rule 1.13.11](#).

- 1.17.6. Some agendas and other cards have abilities with a trigger condition related to an agenda being scored. These abilities become pending after the Corp moves the agenda from its current zone to their score area.
- 1.17.7. Some agendas and other cards have abilities with a trigger condition related to an agenda being stolen. These abilities become pending after the Runner moves the agenda from its current zone to their score area.
- 1.17.8. If an ability meets its trigger condition from an agenda being scored or stolen, the agenda's last known number of advancement **counters** before being scored or stolen is used for any references to the number of advancement **counters** in the effects of that ability.

1.18. Advancing Cards

- 1.18.1. To ADVANCE a card is to place an advancement **counter** from the bank on it. The Corp typically advances cards with a basic action during their action phase, but card abilities can also advance cards.
- 1.18.2. Resolving an instruction that directly places an advancement **counter** onto a card is not the same as advancing a card. Likewise, resolving an instruction that moves an advancement **counter** from one card to another is not the same as advancing a card.
*Example: The Corp plays Mushin-no-Shin, choosing to install an Oaktown Renovation. Because the three advancement **counters** were placed on the card directly, the Corp does not gain any credits from the ability on Oaktown Renovation.*
- 1.18.3. **The Corp can only advance certain installed cards.** Agendas can always be advanced. If a card other than an agenda says that it “can be advanced” or that “you can advance” that card, the card can be advanced even while it is unrezzed. There is no limit to the number of times a card can be advanced.
- 1.18.4. An “advanced card” is a card with 1 or more advancement **counters** hosted on it.

1.19. Trashing

- 1.19.1. TRASHING is the act of moving **an object** to its owner's discard pile.
- 1.19.2.** During access, the Runner has the opportunity to pay the trash cost of the accessed card to trash it. See [section 7.1.5](#).
- 1.19.3.** A trashed card is not considered to have been discarded, and vice versa. Cards that prevent a card from being trashed cannot prevent a card from being discarded.
- 1.19.4.** The symbol [trash] represents trashing the **object** it appears on. It is used as a cost for certain abilities.

Example: Fall Guy's last ability reads “[trash]: Gain 2[c].” The cost to trigger this ability is to trash Fall Guy.

- a. “A [trash] ability” refers to a paid ability that includes [trash] in its trigger cost.

1.20. Memory

- 1.20.1. A MEMORY UNIT (MU) is a space available to the Runner to install programs. Memory units always appear with a quantity, such as 2[MU], which means “2 memory units.”
- 1.20.2. The Runner’s MEMORY LIMIT is the total number of memory units the Runner has. If the Runner’s identity card does not specify a starting memory limit, their starting memory limit is 4[MU]. Card abilities that indicate +[MU] apply that increase to the Runner’s memory limit.

Example: The Runner has Astrolabe installed, which has the ability “+1[MU]”. The Runner can therefore have up to 5[MU] worth of programs installed.

- 1.20.3. Programs have a MEMORY COST that is continually applied against the memory limit. [Rule 3.9.3](#) details how the Runner’s installed programs are constrained by the memory limit.

1.21. Card Visibility

- 1.21.1. Cards in the play area and other public zones are either FACEUP or FACEDOWN. Faceup cards are freely visible to all players. Facedown cards are oriented so that the face containing the card’s information is not visible.
 - a. Cards in hidden or secret zones (see [section 4.1](#)) are neither faceup nor facedown.
- 1.21.2. When a player LOOKS at a card, they are allowed to see the front face of the card even if the front face would not normally be visible to them. A player should look at that card without showing it to the other player if it is not normally visible to that player.
 - a. A player may look at facedown cards they control at any time.
- 1.21.3. To REVEAL a card is to show its front face to all players, then return it to its previous state.
- 1.21.4. Some abilities instruct the Runner to EXPOSE one or more cards. To expose a card is to reveal it, except that only installed, unrezzed cards can be exposed.
- 1.21.5. Effects that instruct a player to look at, reveal, expose, or access a card are not the same, even if they would be performed similarly at times. See [section 7](#) for more information on accessing.
- 1.21.6. If a resolving ability directs one or both players to look at or reveal a card or set of cards, each such card remains visible to the relevant player(s) until the entire ability is finished resolving or the card moves to a different location.
- 1.21.7. The ability “While the Runner is accessing this card from R&D, they must reveal it.” is a static ability that applies for the entire duration of the access. This is also the case for

cards with the older wording, “If [this card] is accessed from R&D, the Runner must reveal it.”

2. Parts of a Card

2.1. Name

- 2.1.1. The identifier of a card is its NAME, alternatively referred to as its title. It is written along the top of the card for identities, agendas, ice, operations, events, and resources, or immediately below the art box in the case of assets, upgrades, hardware, and programs.
- 2.1.2. Identities have a subtitle written immediately below the main title line. Their full name is the combination of their main title and subtitle.
- 2.1.3. When a card refers to its own name, it is a self-reference. See [section 10.11](#).
- 2.1.4. When a card refers to “copies of” cards with a particular name, it refers to any card with that name.

2.2. Unique Symbol

- 2.2.1. Some cards are UNIQUE, and have a unique symbol (◆) before their name to designate this. See [rule 10.1.1](#).

2.3. Play Cost, Install Cost, or Rez Cost

- 2.3.1. The play, install, or rez cost of a card is a number that appears in the top-left corner of most card types. Play and install costs appear within a large circular icon, while rez costs appear within a small octagonal icon.
- 2.3.2. Play, install, and rez costs on cards are all paid in credits. [Section 8](#) details how cards are played, installed, and rezzed.
- 2.3.3. Events and operations have a play cost.
- 2.3.4. Hardware, programs, and resources have an install cost.
 - a. Ice also has an install cost, but it does not appear on the card. The install cost of ice is determined by the configuration of other installed ice rather than being intrinsic to the card. See [rule 1.16.5](#).
- 2.3.5. Assets, ice, and upgrades have a rez cost.

2.4. Advancement Requirement

- 2.4.1. The advancement requirement appears only on agendas. It is a number in the top-right corner of the card, within a small circle.

- 2.4.2. [Section 1.17](#) details how agendas are scored and the purpose of the advancement requirement.

2.5. Agenda Points

- 2.5.1. The agenda point value appears only on agendas. It is the number in the center-left of the card, on top of a graphic depicting three pillars.
- 2.5.2. The agenda points in a player's score area contribute toward that player winning the game. See [section 1.7](#).

2.6. Trash Cost

- 2.6.1. Assets, upgrades, and some ice and operations have trash costs. A trash cost is a number appearing in the lower-right corner of the card within a [trash] symbol.
- 2.6.2. [Section 7.1.5](#) details how the Runner can trash a card they access by paying its trash cost.

2.7. Strength

- 2.7.1. Strength is a number appearing on ice and on **icebreaker** programs. It appears on the bottom-left corner of the card within a large semicircle.
- 2.7.2. Non-**icebreaker** programs have a dash (–) in the bottom-left corner instead of a strength value.
- 2.7.3. [Section 3.9.5](#) discusses the interaction between ice strength and **icebreaker** strength.

2.8. Memory Cost

- 2.8.1. Memory Cost is a number appearing only on programs. It appears to the right of the install cost, within an [MU] symbol.
- 2.8.2. [Section 1.20](#) discusses the topic of memory, and [rule 3.9.3](#) details how programs interact with the Runner's memory limit based on their memory cost.

2.9. Base Link

- 2.9.1. Base link is a number appearing in the top-left of Runner identity cards.
- 2.9.2. The Runner's base link contributes to their link value. [Section 10.7](#) defines how the Runner's link value is calculated, and [section 10.8](#) explains how it contributes to contesting trace attempts.

2.10. Starting Memory Limit

- 2.10.1. Starting memory limit is a number appearing near the top-left of Runner identity cards, within the [MU] symbol.
- 2.10.2. The Runner's memory limit is checked against the total memory cost of programs they have installed. This value is calculated by applying any active modifiers to the Runner's starting memory limit. See [section 1.20](#).

2.11. Minimum Deck Size

- 2.11.1. Minimum deck size is a number shown as the uppermost of two small boxes in the lower-left of Corp identities and the lower-right of Runner identities.
- 2.11.2. [Section 1.4](#) details the rules for building a deck, including the requirement to adhere to the minimum deck size of the deck's associated identity.

2.12. Influence Limit

- 2.12.1. The influence limit is a number shown as the lowermost of two small boxes in the lower-left of Corp identities and the lower-right of Runner identities.
- 2.12.2. [Section 1.4](#) details the rules for building a deck, including the requirement that out-of-faction cards in the deck adhere to the influence limit of the associated identity.

2.13. Faction Affiliation

- 2.13.1. Cards are divided into ten factions: four Corp factions, three Runner factions, and three Runner mini-factions. Factions and influence restrict deckbuilding options, allowing each faction to have distinct play differences from one another.
 - a. The Corp factions are Haas-Bioroid, Jinteki, NBN, and the Weyland Consortium.
 - b. The major Runner factions are Anarch, Criminal, and Shaper.
 - c. The Runner mini-factions are Adam, Apex, and Sunny Lebeau.
- 2.13.2. A card's faction can be identified by the color of its background, as well as a faction logo in one of its four corners. This logo also appears as a watermark behind the text box. If a card has a white background and no logo, it is neutral.
- 2.13.3. Each identity is associated with one faction. [Section 1.4](#) details how factions affect deckbuilding.

2.14. Influence Cost

- 2.14.1. A card's influence cost, or influence value, is represented in a bar with 5 circular slots near the bottom of most cards. The influence cost is the number of slots that are filled with a blue circle, and can range from 0 to 5.

- 2.14.2. Some cards cannot be played out of faction, and therefore do not have an influence value, which is different from an influence cost of “0”. These cards do not have the influence bar at all.
- 2.14.3. Some Corp cards have the subtype **alliance** and text defining adjustments to their influence costs. The ability adjusting the influence cost of an **alliance** card applies only when determining the legality of an already constructed deck and does not have any effect during gameplay. The ability references other cards included in the deck to determine whether the card should apply its printed influence cost or 0 against the deck’s influence limit.
- 2.14.4. [Section 1.4](#) details how a card’s influence cost (or lack thereof) affects deckbuilding.

2.15. Card Type

- 2.15.1. A card’s types appear as a line of text across the center of non-ice cards above the text box. For ice, they run along the left side of the text box, rotated 90° from the rest of the text.
- 2.15.2. Each card has one card type, which can be identity, agenda, asset, ice, operation, upgrade, event, hardware, program, or resource. It is written in bold at the beginning of the type line.
- 2.15.3. While inactive, a card is still considered a card of the type printed on it, with the exception of facedown installed Runner cards. See [section 8.1.4](#).
- 2.15.4. Abilities can change a card from its printed type to an agenda. See [rule 10.1.3](#).
- 2.15.5. Counters never have card types.

2.16. Subtypes

- 2.16.1. Cards may have one or more subtypes, which are written following the primary type. Subtypes are card descriptors that can be referenced by card abilities. Cards that share a subtype often share a distinct characteristic or ability.
 - a. The subtypes **region**, **console**, and **icebreaker** have special rules, discussed in sections [3.6.5](#), [3.8.5](#), and [3.9.5](#) respectively. Other subtypes have no intrinsic rules meaning.
- 2.16.2. In card text, subtypes are always referenced in bold text. If a word or phrase that is also a subtype appears in a card’s text but is not in bold text, that word or phrase does not reference the subtype.
- 2.16.3. Some abilities reference cards by a lack of subtype, such as “non-**virus**”. These abilities can only affect cards that do not have the specified subtype, regardless of any other subtypes the cards do have.

- 2.16.4. A card has all of its subtypes at all times, even while inactive, with the exception of facedown installed Runner cards. See [section 8.1.4](#).
- 2.16.5. Abilities can add or remove subtypes. The number of times a card has, gains, and loses each subtype is tracked as a running total, but the result is binary: if the total number of instances of the subtype exceeds the number of times it is removed, the card has that subtype. Otherwise, it does not.

*Example: The Runner plays Tinkering on a Wendigo hosting 1 advancement **counter**. Wendigo has two instances of **code gate**, one printed and one from Tinkering, and has lost one instance of **code gate** from its own ability. Thus it is still a **code gate**.*

- 2.16.6. If an effect requires a player to choose a subtype, that player must choose a single subtype that has been printed on at least one card. If a player is required to choose a subtype belonging to a specific card type, they must choose a subtype that has been printed on at least one card with that type.
- 2.16.7. Below is a list of all subtypes.
- a. The identity subtypes are Bioroid, Clone, Corp, Corporation, Cyborg, Digital, Division, G-mod, Megacorp, Natural, Police Department, Stealth, and Subsidiary.
 - b. The agenda subtypes are Ambush, Expansion, Initiative, NEXT, Psi, Public, Research, Security, Sensie, and Source.
 - c. The asset subtypes are Advertisement, Alliance, Ambush, Beanstalk, Bioroid, Cast, Character, Clone, Corporation, Executive, Facility, Government, Hostile, Illicit, Political, Psi, Region, Research, Ritzy, Seedy, and Transaction.
 - d. The ice subtypes are AP, Advertisement, Ambush, Barrier, Bioroid, Code Gate, Deflector, Destroyer, Grail, Illicit, Morph, Mythic, NEXT, Observer, Psi, Sentry, Tracer, and Trap.
 - e. The operation subtypes are Alliance, Black Ops, Condition, Current, Double, Gray Ops, Illicit, Lockdown, Psi, Reprisal, Terminal, Transaction, and Triple.
 - f.** The upgrade subtypes are Advertisement, Alliance, Ambush, Beanstalk, Bioroid, Character, Clone, Enforcer, Executive, Facility, Hostile, Off-site, Orgcrime, Psi, Region, Ritzy, Seedy, Security Protocol, Sysop, and Unorthodox.
 - g. The event subtypes are Condition, Current, Double, Job, Mod, Orgcrime, Priority, Run, Sabotage, Stealth, and Terminal.
 - h. The hardware subtypes are Bomb, Chip, Companion, Console, Consumer-grade, Cybernetic, Gear, Link, Mod, Stealth, Vehicle, and Weapon.
 - i. The program subtypes are AI, Caïssa, Cloud, Daemon, Decoder, Deva, Fracter, Icebreaker, Killer, Stealth, and Virus.

- j. The resource subtypes are Clan, Companion, Connection, Directive, Genetics, Government, Job, Link, Location, Remote, Ritzzy, Sabotage, Seedy, Source, Stealth, and Virtual.

2.17. Text Box

- 2.17.1. The TEXT BOX is the central area on a card containing any abilities the card has. Abilities usually only apply when the card they appear on is active. [Rule 9.1.8](#) details exceptions.
- 2.17.2. Many cards have italicised text written after their game text. This is FLAVOR TEXT, and has no effect on the game or the function of the card, nor can other abilities reference or interact with it.
- 2.17.3. Some cards have italicised text in parenthesis. This is REMINDER TEXT, and has no effect on the game or the function of the card, nor can other abilities reference or interact with it directly. Reminder text simply summarizes a rule that may apply to that card text.

3. Card Types

3.1. Identities

- 3.1.1. Each player starts the game with a single identity active in the play area. See [section 1.6](#).
 - a. Some identities are double-sided. Only the front side of a double-sided identity will have values for its minimum deck size and influence limit. Only the side of a double-sided identity that is currently faceup is active at any given time.
 - b. Identities are not installed.
- 3.1.2. Only identity cards have minimum deck sizes and influence limits. These values do not apply during the game; instead, they determine deckbuilding constraints for decks associated with that identity. See [section 1.4](#).
- 3.1.3. Only Runner identities have a base link. This value is used in calculating the Runner's link value. See [section 10.7](#).
- 3.1.4. The Corp's identity card is used to mark the location of HQ. See [section 4.3.8](#).

3.2. Agendas

Agendas represent valuable pieces of Corporate data. When the Corp advances and scores agendas, it represents completion of critical Corporate projects. When the Runner steals agendas, it represents their acquisition of secret Corporate data.

- 3.2.1. The Corp may install agendas only in **the roots of** remote servers. There can be only one agenda or one asset installed in **the root of** any given remote server at a time. See [section 4.6.8](#).
- 3.2.2. Only agendas have advancement requirements and agenda points. See [section 1.17](#).
 - a. An agenda's advancement requirement is the minimum number of advancement **counters** that must be on that agenda for the Corp to score it.
 - b. The agenda points on an agenda contribute to a player's score while the agenda is in their score area.
- 3.2.3. Agendas cannot be rezzed and are inactive while installed. Agendas are active in the Corp's score area. Agendas are inactive in the Runner's score area unless the agenda's card text specifies otherwise. See [section 4.5](#).
 - a. Some agendas may be installed faceup or turned faceup through card abilities. Faceup agendas are neither rezzed nor unrezzed. If an agenda's printed text directs the Corp to install it faceup, that agenda's abilities are active while it is installed.
 - b. Some agendas modify the requirements for stealing or scoring them. These abilities are always active, according to any restrictions or requirements specified. See [section 9.1.8](#).
- 3.2.4. Agendas can always be advanced. See [section 1.18](#).

3.3. Assets

Assets represent various resources and connections at the Corp's disposal.

- 3.3.1. The Corp may install assets only in **the roots of** remote servers. There can be only one agenda or one asset installed in **the root of** any given remote server at a time. See [section 4.6.8](#).
- 3.3.2. Assets are among the card types that have rez costs. They are installed unrezzed and remain inactive until the Corp rezzes them by paying their rez costs during a paid ability window. See [section 8.1](#).
- 3.3.3. Assets are among the card types that have trash costs. If the Runner accesses an asset, they may pay the trash cost to trash that asset. See [section 7.1.5](#).
- 3.3.4. One asset (*Lady Liberty*) has the subtype **region** and the text "Limit 1 **region** per server." See [section 3.6.5](#) for rules governing **region** cards.

3.4. Ice

Ice are sophisticated security systems that protect Corporate servers through a variety of means. “Ice” has been suggested to stand for a variety of expressions, but is most commonly expanded as “Intrusion Countermeasures Electronics”.

- 3.4.1. “Ice” and “piece of ice” are synonymous.
- 3.4.2. Ice are the only type of card installed protecting servers. Ice are not installed in **the roots of** servers. See [section 4.6.9](#).
 - a. Ice protecting a server are laid out horizontally in front of that server, with the outermost piece of ice furthest from the server.
 - b. When the Corp installs a piece of ice, they must install it in the outermost position protecting a server. See [section 8.5](#).
- 3.4.3. Ice are among the card types that have rez costs. They are installed unrezzed and remain inactive until the Corp rezzes them by paying their rez costs. See [section 8.1](#).
 - a. Unlike assets and upgrades, ice can normally only be rezzed while being approached during a run. See [section 6.4](#).
- 3.4.4. Ice are among the card types that have strength. An ice’s strength determines which **icebreakers** are allowed to **interface** with it. See [section 3.9.5](#).
- 3.4.5. Some ice have trash costs. If the Runner accesses an ice with a trash cost, they may pay the trash cost to trash that ice. See [section 7.1.5](#).
 - a. Unlike assets, upgrades, and agendas, ice are not normally accessed while installed. Encountering a piece of ice does not entail accessing that ice and does not give the Runner an opportunity to pay its trash cost.
- 3.4.6. Ice are the only Corp cards with install costs. The install cost of a piece of ice is not a printed value but instead a calculated value based on the number of other ice already protecting the relevant server. See [section 1.16.5](#).
- 3.4.7. Ice are the only cards with subroutines. **See** [section 9.8](#).

3.5. Operations

- 3.5.1. Operations are the only Corp cards that are played. Operations are never installed. Playing an operation causes its abilities to become active, and once those abilities are fully resolved the operation is trashed. See [section 8.6](#).
 - a. Some operations have the subtype **condition**. As part of their abilities, **condition** operations convert themselves into condition counters and install themselves. A **condition** operation cannot be installed by an install effect; it can only be installed through its own conversion ability. See [rule 10.1.4](#).

- b. Some operations have the subtype **current**. As part of their abilities, **current** operations prevent themselves from being trashed until another **current** operation or event is played or the Runner steals an agenda. When one of these occurs, the active **current** is trashed during the next checkpoint. See [section 10.3](#).
 - c. Some operations have the subtype **lockdown**. As part of their abilities, **lockdown** operations prevent themselves from being trashed until the start of the Corp's next turn after the operation is played. Once the Corp's turn begins, this effect expires and the operation is trashed during the next checkpoint, before any conditional abilities marked pending at that time can be triggered. See [section 10.3](#).
- 3.5.2. Operations are the only Corp cards with play costs. An operation's play cost is the number of credits the Corp must spend to play the operation. See [section 8.6](#).
- 3.5.3. Some operations have trash costs. If the Runner accesses an operation with a trash cost, they may pay the trash cost to trash that operation. See [section 7.1.5](#).

3.6. Upgrades

- 3.6.1. The Corp may install upgrades **in the root of any server**. Upgrades are the only card type that can be installed in the root of a central server.
- 3.6.2. There is no limit to the number of upgrades that can be installed in **the root of a single server**.
- 3.6.3. Upgrades are among the card types that have rez costs. They are installed unrezzed and remain inactive until the Corp rezzes them by paying their rez costs during a paid ability window. See [section 8.1](#).
- 3.6.4. Upgrades are among the card types that have trash costs. If the Runner accesses an upgrade, they may pay the trash cost to trash that upgrade. See [section 7.1.5](#).
- 3.6.5. Regions
- a. Some upgrades have the subtype **region**. All **regions** have the text "Limit 1 **region** per server."
 - b. The **region** limitation applies only to cards with the subtype **region** that are installed or in the process of being installed **in the root of a server**. It does not apply to cards in central servers, and it does not apply to ice.
 - c. The **region** limitation applies to all cards as specified in [rule 3.6.5b](#), regardless of the active or inactive state of those cards.
 - d. When the Corp installs a card with the subtype **region** into **the root of** a server that already has a **region**, they must trash the already-installed **region**. See [section 8.5](#).

- e. The Corp cannot swap a **region** card with a non-**region** card in another location if there is another **region** card already installed in the destination location.

3.7. Events

- 3.7.1. Events are the only Runner cards that are played. Events are never installed. Playing an event causes its abilities to become active, and once those abilities are fully resolved the event is trashed. See [section 8.6](#).
 - a. One event, *On the Lam*, has the subtype **condition**. As part of its abilities, it converts itself into a condition counter and installs itself. This event cannot be installed by an install effect; it can only be installed through its own conversion ability. See [rule 10.1.4](#).
 - b. Some events have the subtype **current**. As part of their abilities, **current** events prevent themselves from being trashed until another **current** operation or event is played or the Corp scores an agenda. When one of these occurs, the active **current** is trashed during the next checkpoint. See [section 10.3](#).
- 3.7.2. Events are the only Runner cards with play costs. An event's play cost is the number of credits the Runner must spend to play the event. See [section 8.6](#).

3.8. Hardware

The Runner's hardware are the physical tools at their disposal: the computers, chips, and other machinery that make up their rig.

- 3.8.1. "Hardware" and "piece of hardware" are synonymous.
- 3.8.2. The Runner installs hardware into the play area.
- 3.8.3. Hardware are among the types of cards with install costs. The Runner installs a hardware faceup and active by paying its install cost. See [section 8.5](#).
- 3.8.4. There is no limit to the number of hardware the Runner can have installed.
- 3.8.5. Consoles
 - a. Some hardware have the subtype **console**. All **consoles** have the text "Limit 1 **console** per player."
 - b. **If a player ever controls more than one installed console, all but the most recently installed console are trashed. Trashing cards this way cannot be prevented.**
 - c. This limitation applies only to hardware with the subtype **console**.
 - d. This limitation applies only to active **consoles**. A player can have multiple copies of a **console**, and even different **consoles**, in their deck..

3.9. Programs

Programs are digital tools at the Runner's disposal, primarily used as a means of intrusion.

- 3.9.1. The Runner installs programs into the play area.
- 3.9.2. Programs are among the types of cards with install costs. The Runner installs a program faceup and active by paying its install cost. See [section 8.5](#).
- 3.9.3. Programs are the only card type that have a memory cost. A program's memory cost is the number of memory units that program requires while it is installed.
 - a. **Installing a program does not permanently consume memory. The program claims its designated quantity of memory units until it becomes uninstalled, at which point those memory units become available for use by other programs.**
 - b. When the Runner installs a program that would increase the total memory cost of installed programs over their memory limit, they must trash one or more installed programs such that the total memory cost of installed programs including the new program will not exceed their memory limit. See [section 8.5](#).
 - c. If the Runner's memory limit decreases below the total memory cost of installed programs, or if the total memory cost of installed programs increases above the memory limit for any reason other than installing another program, the Runner must trash one or more installed programs such that the total memory cost of the remaining programs does not exceed their memory limit. This occurs during the next checkpoint after the memory limit is exceeded. See [section 10.3](#).
 - d. A program's memory cost is not a cost. It cannot be increased, lowered, ignored, or otherwise modified by abilities that affect costs to install programs.
- 3.9.4. Programs are among the card types that have strength.
 - a. Programs with the subtype **icebreaker** have a printed numerical strength value. An **icebreaker's** strength determines which ice it is allowed to **interface** with. See [section 3.9.5](#).
 - b. Programs without the subtype **icebreaker** do not have a strength value. A dash (–) is printed on non-**icebreaker** program cards in the place where strength is printed on **icebreaker** program cards.
- 3.9.5. Icebreakers
 - a. Some programs have the subtype **icebreaker**. **Icebreaker** programs have strength and abilities that **interface** with ice.
 - b. Paid abilities on an **icebreaker** that modify that **icebreaker's** strength implicitly have a duration of “for the remainder of the current encounter”.

Example: Corroder has the ability “1[c]: +1 strength.” Since no duration is specified, this strength increase applies until the end of the current encounter.

- c. If an **icebreaker**'s paid ability specifies another duration for modifying its strength, that modification lasts until both the stated duration and the implicit encounter duration have expired.

Example: Gordian Blade has the ability “1[c]: +1 strength for the remainder of this run.” This ability applies until the end of the current run, whether it is triggered during an encounter or another step of the run. If the ability is triggered during an encounter that takes place outside of a run, the strength increase applies for the duration of that encounter.

- d. If an **icebreaker**'s paid ability modifies its strength outside of an encounter and does not specify another applicable duration, the modification expires during the next checkpoint.

Example: If the Runner uses Corroder's ability to increase its strength outside of an encounter, it applies only until the next checkpoint.

- e. All **icebreakers** have abilities that **INTERFACE with ice, sometimes referred to as “interface abilities”**. **These abilities are indicated by the “interface” flag preceding the ability's text.**
- f. The Runner can only use an **interface ability** during [step 6.9.3b](#) of an encounter, **and can only choose the encountered piece of ice and its subroutines as targets for the ability's effects. See [section 1.15](#) for more information about targeting.**
- g. The Runner can only use an **interface** ability on an **icebreaker** if the **icebreaker** has strength greater than or equal to the strength of the **encountered** ice.
- h. If an **interface ability** specifies that it affects a particular subtype of ice, then that ability can only be used on a piece of ice that has the specified subtype. If an ability on an **icebreaker** does not specify a subtype of ice that it affects, that ability can be used on any piece of ice.

3.10. Resources

Resources are a wide variety of connections and skills that aid the Runner.

- 3.10.1. The Runner installs resources into the play area.
- 3.10.2. Resources are among the types of cards with install costs. The Runner installs a resource faceup and active by paying its install cost. See [section 8.5](#).
- 3.10.3. There is no limit to the number of resources the Runner can have installed.
- 3.10.4. While the Runner is tagged, the Corp may spend [click] and 2[c] to trash an installed resource as a basic action. See [section 10.5](#).

4. Game Zones

4.1. General

- 4.1.1. A ZONE is a place in which cards **and counters** can exist during the game.
- There are **8** zone types: deck, hand, discard pile, score area, play area, **bank**, set aside, and removed from the game.
 - Each player has their own deck, hand, discard pile, and score area. All players share the play area, **the bank**, the set-aside zone, and the removed-from-game zone.
 - Each card in the game exists in exactly 1 zone at a time.
- 4.1.2. Game rules and card effects can move cards **and counters** between zones. **See [section 8.2](#) for rules on card movements.**
- If a facedown card or a card in a hidden zone would ever move to another zone as part of an effect that requires that card to have a specific attribute, that card must be revealed before being moved to the new zone. **This does not apply to cards that are** being installed.
- Example: When resolving Clone Suffrage Movement, the Corp chooses a facedown Fast Break in Archives. The Corp must reveal the Fast Break before adding it to HQ, to demonstrate that it is indeed an operation, meeting Clone Suffrage Movement's requirement.*
- When a card is moved from one zone to another, the move is instantaneous. The card is never in both zones at once, and it is never not in any zone at all.
- 4.1.3. A card that is not in any zone is outside the game. Cards outside of the game are distinct from cards in the removed-from-game zone, and cannot affect the game. They can only be affected by abilities that explicitly bring in extra cards from outside the game. See [section 1.5](#).
- 4.1.4. A PUBLIC zone is one **where cards are freely visible to both players unless they are facedown**. The play area, **the bank**, the set-aside zone, the removed-from-game zone, and both players' score areas and discard piles are public zones. **See [section 1.21](#) for rules on card visibility and [section 8.1](#) for rules on the faceup and facedown status of a card.**
- 4.1.5. A HIDDEN zone is one where cards are not visible to either player. Each player's deck is a hidden zone.
- 4.1.6. A SECRET zone is one where cards are visible only to their controller. Each player's hand is a secret zone.

- 4.1.7. A LOCATION is a specific position within a zone that **an object** can occupy. A DESTINATION is a location to which **an object** is to be moved.
- a. A card in a player's deck has a location identified by how many cards it is from the top of that deck.
 - b. **Some objects in the play area are in specific locations. See [section 4.6.5](#).**
 - c. The location of a hosted **object** is its host card.
 - d. All other **objects** are simply located in their zone and do not have a more specific location.

4.2. Deck

- 4.2.1. When the game begins, each player's deck presented during setup becomes their in-game draw deck.
- 4.2.2. Decks must be kept facedown with their cards hidden from both players. Neither player may look at the cards in any deck without being explicitly directed to do so by a rule or card ability.
- 4.2.3. **Decks are ordered. The location of each card in a deck must be maintained except when a player is explicitly directed to manipulate the cards in a deck by a rule or card ability. See rule [4.1.7a](#).**
- 4.2.4. The number of cards in a deck is open information. Any player may count the number of cards in a deck at any time.
- 4.2.5. Cards in a deck are inactive.
- 4.2.6. Each player's deck has its own name as a means of differentiating between the two.
- 4.2.7. R&D

R&D stands for Research and Development. It represents upcoming projects and acquisitions the Corp will soon have at their disposal.

- a. R&D is the name for the Corp's deck. It is also one of the Corp's central servers. See [section 4.6.7](#).
- b. If the Corp must ever draw from R&D while it is empty, the Runner wins the game. See [section 1.7](#).

4.2.8. Stack

The stack represents the Runner's untapped potential: future plans, connections, and opportunities they have yet to discover.

- a. The STACK is the name for the Runner's deck.

4.3. Hand

- 4.3.1. A player's HAND is the zone in which they hold the cards they have drawn from their deck. Cards can be added to a player's hand by other rules and card abilities as well. During setup, each player draws a starting hand. See [section 1.6](#).
- 4.3.2. A player's hand is kept secret from their opponent. A player may look at the cards in their own hand, but not at any of the cards in their opponent's hands.
- 4.3.3. Hands are not ordered. A player may freely arrange the cards in their hand in any order at any time.
- 4.3.4. The number of cards in a hand is open information to all players. Any player may count the number of cards in a hand at any time.
- 4.3.5. Cards in a hand are inactive.
- 4.3.6. Each player has a maximum hand size, usually five. A player may have any number of cards in their hand at any given time, but during the discard phase of a player's turn they must discard cards from their hand until there are no more cards in their hand than their maximum hand size allows. See [section 5.5](#).
- 4.3.7. Each player's hand has its own name as a means of differentiating between the two.
- 4.3.8. HQ

HQ stands for Headquarters. It represents the projects, staff, technology, and other assets the Corp is authorized to deploy.

- a. HQ is the name for the Corp's hand. It is also one of the Corp's central servers. See [section 4.6.7](#).
- b. The Corp's identity card represents their HQ server for the purposes of gameplay. Ice protecting HQ is installed in front of the Corp's identity card and upgrades installed in the root of HQ are installed behind the Corp's identity card.

4.3.9. Grip

The grip represents the Runner's currently accessible thoughts, plans, knowledge, potential social connections, and other intangible information that can be leveraged.

- a. The GRIP is the name for the Runner's hand.
- b. Suffering damage forces the Runner to discard cards from their grip at random, and suffering brain damage also reduces the Runner's maximum hand size. The Runner flatlines and loses the game if they suffer more damage than they have cards in grip or must discard down to a hand size lower than zero. See [section 10.4](#).

4.4. Discard Pile

- 4.4.1. A player's DISCARD PILE is the zone in which they place cards that have been trashed or discarded. Discard piles start the game empty.
- 4.4.2. Discard piles are not ordered. A player may freely arrange the cards in their discard pile in any order at any time.
- 4.4.3. The number of cards in a discard pile is open information. Any player may count the number of cards in a discard pile at any time.
- 4.4.4. Cards in a discard pile are inactive.
- 4.4.5. Each player's discard pile has its own name as a means of differentiating between the two.
- 4.4.6. Archives

Archives represents the Corp's internal record database of all scrapped, abandoned, or discontinued projects, contracts, assignments, and other endeavors.

- a. ARCHIVES is the name for the Corp's discard pile. It is also one of the Corp's central servers. See [section 4.6.7](#).
- b. Cards in Archives can be either faceup or facedown. If a Corp card is visible to the Runner when it is trashed or discarded, it is put in Archives faceup. If a Corp card is not visible to the Runner when it is trashed or discarded, then it is put in Archives facedown. Facedown cards in Archives should be oriented horizontally so that the Runner can easily see they are present.
- c. The faceup cards in Archives are open information. Any player may look through the faceup cards in Archives at any time. The facedown cards in Archives are visible only to the Corp. The Corp may look through the facedown cards in Archives at any time, but the Runner cannot look at them.

4.4.7. Heap

The heap represents what the Runner has lost: their old jobs, favours burned, friends long gone, gear wrecked, and code erased.

- a. The HEAP is the name for the Runner's discard pile.
- b. The cards in the heap are open information. Any player may look through the cards in the heap at any time.

4.5. Score Area

The score areas represent major accomplishments for each player in service of their long-term ambitions. The Corp's score area contains all large-scale projects they have seen to completion. The

Runner's score area is a memorial to all such initiatives they have successfully disrupted, exposed, or sold details of to the highest bidder.

- 4.5.1. A player's SCORE AREA is the zone in which they place agendas they have scored or stolen. [Section 1.17](#) details scoring and stealing agendas. Score areas start the game empty.
- 4.5.2. The agendas in a score area are open information. A player may look through or count the agendas in a score area at any time.
- 4.5.3. Score areas are not ordered. A player may freely arrange the agendas in their score area in any order at any time.
- 4.5.4. Agendas in the Corp's score area are active. Agendas in the Runner's score area are inactive unless stated otherwise.

4.6. Play Area

- 4.6.1. The PLAY AREA is the shared active zone into which cards are played or installed. The play area normally starts the game empty except for each of the player's identities.
 - a. [Section 1.5](#) details cases where an ability may cause cards to start the game in the play area.
- 4.6.2. The number of cards in the play area and where in the play area they are located is always open information.
- 4.6.3. Whether or not a card itself and its attributes are open information while in the play area is determined by the card's status. Cards that are faceup are open information and can be examined by any player at any time. Cards that are facedown are secret information and can only be examined by the player who controls the card. See [section 10.2](#).
- 4.6.4. Whether or not a card is active while in the play area is determined by the card's status.
 - a. Identities start the game in the play area and are always active.
 - b. Generally, Corp agendas, assets, ice, and upgrades are installed into the play area unrezzed and thus inactive. They are active while rezzed. See [section 8.5](#).
 - c. Generally, Runner hardware, programs, and resources are installed into the play area faceup and active. See [section 8.5](#).
 - d. Some Runner cards are installed facedown or turned facedown through card abilities. This is distinct from the status of rezzed/unrezzed that Corp cards can have. Facedown Runner cards are not active. See [section 8.1.4](#).
 - e. Operations and events are played into the play area faceup and are active for the duration of their resolution before being trashed. See [section 8.6](#).
- 4.6.5. Some **objects** in the play area are organized into specific orientations and/or locations.

- a. Corp cards are arranged into distinct servers. Agendas and assets are installed in **the roots of** remote servers, upgrades are installed in **the roots of any servers**, and **pieces of** ice are installed horizontally protecting servers. See [section 4.6.6](#) and [section 4.6.9](#).
- b. The Corp's identity is used to indicate where in the play area HQ exists for the purposes of installing upgrades in the root of HQ and ice protecting HQ. See [section 4.6.7](#).
- c. The Runner's identity, hardware, programs, and resources do not have specific locations in the play area.
- d. There is no specific location in the play area for operations or events. They are simply played into the play area, where they resolve before being trashed. See [section 8.6](#).
- e. **Each player has a credit pool, which is a location in the play area where credits they control are kept.**
- f. **Counters other than credits have no specific location in the play area. Tag counters are always controlled by the Runner, and bad publicity counters are always controlled by the Corp. Other types of counters are either nonfunctional gameplay aids or appear only in host relationships. See also [section 1.9](#).**
- g. Some cards can act as install locations for other cards while in the play area, such that cards of a specific type can be hosted directly onto them during the installation process. In this type of host relationship, both the host card and the hosted card are in the play area, but the host relationship can affect certain aspects of either the card or the installation process. See [section 1.13](#).
- h. Some cards can host other cards without installing them. In this type of host relationship, both the host card and the hosted card are in the play area, but the hosted card is not installed and thus not active. See [section 1.13](#).

4.6.6. Servers

- a. A SERVER is a **set of locations** where the Corp installs their cards into the play area and against which the Runner can initiate runs. Whenever a Corp installs an agenda, asset, ice, or upgrade, they must choose a destination server for that card. See [section 8.5](#).
- b. Once a Corp card is installed, it cannot be moved from its server without explicit direction from a game rule or card effect.
- c. The two types of servers are central servers and remote servers. See sections [4.6.7](#) and [4.6.8](#), respectively.
- d. All servers can have ice installed protecting them. See [section 4.6.9](#).

- e. **All servers have a ROOT where cards can be installed. A central server's root can contain any number of upgrades, while a remote server's root can contain 1 asset or agenda and any number of upgrades. Assets and agendas cannot be installed in the root of a central server.**
- f. **All cards installed in the root of a server are kept in the same orientation and location together. The order in which the cards are installed is open information, but the type of cards installed in the root of a remote server should not be derivable from their orientation or location.**
- g. **Only one region upgrade can be installed in the root of a server at a time. See [section 3.6.5](#).**
- h. **Some upgrades have restrictions limiting them to specific servers. See [rule 8.5.12](#).**

4.6.7. Central Servers

- a. The Corp has three CENTRAL SERVERS at all times. **Each central server corresponds to one of the Corp's zones: HQ corresponds to the Corp's hand, R&D corresponds to the Corp's deck, and Archives corresponds to the Corp's discard pile.**
- b. The Corp's identity is in the play area to indicate the location of HQ relative to other servers, but it is not in HQ, in the root of HQ, or protecting HQ. The cards in HQ are held in the Corp's hand. The upgrades installed in the root of HQ are placed in a vertical orientation **near** the Corp's identity. The ice installed protecting HQ are placed in a horizontal orientation in front of the Corp's identity outwards towards the Runner.
- c. The Corp's deck is placed in the play area to indicate the location of R&D relative to other servers, but the cards in the deck are not installed. The cards in R&D are the cards in the deck. The upgrades installed in the root of R&D are placed in a vertical orientation **near** the deck. The ice installed protecting R&D are placed in a horizontal orientation in front of the deck outwards towards the Runner.
- d. The Corp's discard pile is placed in the play area to indicate the location of Archives relative to other servers, but the cards in the discard pile are not installed. The cards in Archives are the cards in the discard pile. The upgrades installed in the root of Archives are placed in a vertical orientation **near** the discard pile. The ice installed protecting Archives are placed in a horizontal orientation in front of the discard pile outwards towards the Runner.

4.6.8. Remote Servers

- a. **A REMOTE SERVER is the only type of server that can contain an asset or agenda in its root.**

- b. The Corp starts the game with no remote servers. The Corp creates a new remote server by **declaring it as the destination server for a card they are about to install**. There is no limit to the number of remote servers the Corp can have at any given time.
- c. Cards installed **in the roots of** separate remote servers are placed in a vertical orientation in a row extending away from the central servers to indicate the locations of the remote servers relative to other servers. The ice installed protecting each remote server are placed in a horizontal orientation in front of any cards in **the root of** that server, outwards towards the Runner.
- d. A remote server exists as long as there is at least one card **in the root of** or protecting it. A remote server can be made up of only ice protecting it, only cards installed in it, or a combination of both.
- e. During checkpoints, a remote server ceases to exist if there are no cards installed in **its root** and no cards installed protecting it. If a server ceases to exist during a run against it, then the run ends after any currently open paid ability window closes. Unless the run has already been declared either successful or unsuccessful, it is neither. See [rule 8.5.8](#).
- f. The Corp card *Earth Station: SEA Headquarters* and its reverse, *Earth Station: Ascending to Orbit*, both have the ability, “Limit 1 remote server.” While this ability is active, as long as a remote server exists, the Corp cannot create a new remote server. If at any time more than 1 remote server exists while this ability is active, the next checkpoint will force the Corp player to choose 1 of those servers. The checkpoint will then trash all cards in **the root of** and protecting each other remote server. See [step 10.3.1e](#).

4.6.9. Protecting a Server

- a. **An installed piece of ice** is arranged in a horizontal position in front of the server it is protecting. **Ice are ordered within the play area. The location of a piece of ice is defined by** the server it is protecting **and** the number of other pieces of ice between it and the server.
- b. If there is no other ice between the server and a piece of ice, that ice is the innermost piece of ice. If there is no ice farther from the server than a piece of ice, that ice is the outermost piece of ice. If there is only one piece of ice protecting a server, that ice is both the innermost and the outermost piece of ice.
- c. If two pieces of ice protecting two different servers share the same number of ice between them and their respective servers, those pieces of ice are considered to be in the same position.
- d. When the Corp installs a piece of ice protecting a server, they must place it in the outermost position protecting that server. See [section 8.5](#).

- e. If the number of other ice between a piece of ice and the server changes, its position will change accordingly, moving either closer or farther from the server. If this occurs while that ice is being approached, encountered, or passed during a run, the Runner's position attacking the server changes in the same way.

4.7. Bank

- 4.7.1. **The BANK is an inactive zone shared by both players. It holds an unlimited supply of counters for use during gameplay.**
- 4.7.2. **Counters placed or loaded onto a card are taken from the bank unless otherwise specified. Counters gained by players are taken from the bank unless otherwise specified.**
- 4.7.3. **Credits and other counters that are spent to pay costs return to the bank.**
- 4.7.4. **Counters that are not in a legal location return to the bank during checkpoints. See [section 10.3](#).**
- 4.7.5. **Cards are never located in the bank.**

4.8. Set Aside

- 4.8.1. The SET-ASIDE ZONE is an inactive zone shared by both players. It is used as a temporary holding space for **objects** being affected by abilities.
- 4.8.2. If **an object** is "set aside", it is moved from its current zone to the set-aside zone.
- 4.8.3. An ability cannot see the set-aside zone unless it specifically sets **objects** aside or refers to set-aside **objects**. Outside of these cases, cards moved into another zone from the set-aside zone are treated as though they entered that zone directly from their location prior to being set aside.

Example: The Runner is playing as Exile and plays Test Run, searching their heap for a program. Although the program is set aside before being installed, Exile's ability treats it as though it were installed directly from the heap, and so Exile will draw a card.

- 4.8.4. Cards found during a search are set aside facedown while the search completes (in particular, while any deck that was searched is shuffled). See [section 8.7](#).
- 4.8.5. Some paid abilities with a trigger cost that uninstalls or forfeits the ability's source card need to **refer to or** act on that card's hosted cards or counters. These abilities set aside the hosted cards or counters as the trigger cost is paid. See [rule 9.5.5](#).
- 4.8.6. Cards set aside by a card ability are faceup unless the ability specifically instructs that they be set aside facedown.

4.9. Removed from the Game

- 4.9.1. The REMOVED-FROM-GAME ZONE is an inactive zone shared by both players. It is used to take cards out of any player's control until the end of the game.
- 4.9.2. If a card is removed from the game, it is moved from its current zone to the removed-from-game zone.
- 4.9.3. Forfeiting an agenda removes it from the game. See [rule 8.2.5](#).
- 4.9.4. Cards that have been removed from the game are inactive. Their abilities are inactive even if they fall under one of the exceptions that would normally apply to inactive cards.
- 4.9.5.** A card that has been removed from the game cannot move out of the removed-from-game zone or otherwise be interacted with.
- 4.9.6. Cards that have been removed from the game are open information. Any player may look through or count the cards that have been removed from the game at any time.

5. Turns

5.1. General

- 5.1.1. A TURN is a duration of time during which a player may take actions and is the first to receive priority to act during **priority** windows.
- 5.1.2. The Corp's turn consists of three phases. During the DRAW PHASE, the Corp draws the top card of R&D into HQ; during the ACTION PHASE, the Corp spends [click]s to perform actions; during the DISCARD PHASE, the Corp discards until the number of cards in HQ does not exceed their maximum hand size. Once the Corp's turn ends, the Runner's turn begins.
- 5.1.3. The Runner's turn consists of two phases. During the ACTION PHASE, the Runner spends [click]s to perform actions; during the DISCARD PHASE, the Runner discards until the number of cards in their grip does not exceed their maximum hand size. Once the Runner's turn ends, the Corp's turn begins.
- 5.1.4. The beginning and end of each player's turn are formalized as particular steps of the turn timing structures.**
 - a. Trigger conditions related to a turn beginning are not met as the turn timing structure opens. They are met at the timing step that indicates the formal beginning of the turn.**
 - b. Trigger conditions related to a turn or discard phase ending are met at the timing step that indicates the formal end of the turn. They are not met when**

the turn timing structure later closes. The end of the turn and the end of the discard phase are the same.

- c. Some effects can occur before the “turn begins” step or after the “turn ends” step of a turn. The player whose turn timing structure is in progress is still the active player, and effects that apply during their turn are still applicable, even though their turn has not yet formally begun or has already formally ended.
- d. Once the game begins, it is always one or the other player’s turn. There is no time “between turns”.

5.2. Actions

5.2.1. An ACTION is any paid ability **where the cost begins with a [click] symbol.**

- a. Other costs can contain [click] symbols without denoting an action.

Example: The ability “[click]: Gain 1[c] and draw 1 card.” on Professional Contacts is an action. The ability “Lose [click]: Break 1 subroutine on this ice. Only the Runner can use this ability.” on Eli 1.0 is not an action and is used during a paid ability window.

5.2.2. A player initiates an action by paying its cost in [click], as well as any other costs associated with that action, during their action window.

- a. Once an action is initiated, it must be completed before the game can advance to the next step or open another action window. If the effects of an action initiate a timing structure, such as a run, players may use paid abilities, rez cards, and score agendas as dictated by the priority windows of that nested timing structure, but otherwise players cannot perform any of those voluntary effects while the action is resolving. See [Section 9.2](#) for more details about priority.
- b. If a timing structure is initiated during the resolution of an action, that action is not complete until the new timing structure is complete and any further effects of the initiated action following the completion of the new timing structure are resolved.

Example: The Runner takes the action “play an event”, playing Stimhack. The action is not complete until the run initiated by Stimhack ends, the Runner suffers brain damage, and Stimhack is trashed following its resolution.

- c. A conditional ability that meets its trigger condition due to a player taking an action resolves after the action’s trigger cost is paid, before the effects of the action itself. See [rule 9.1.2a](#).

5.2.3. Each player has several basic actions they can always perform. The Corp’s basic actions are listed in [section 5.2.7](#), and the Runner’s basic actions are listed in [section 5.2.8](#).

- 5.2.4. A player can only take an action if its effect has the potential to change the game state. See [rule 1.2.5](#).
- 5.2.5. A player can only take actions during the action phase of their turn. If an effect directs a player to take an action outside of that player's action phase, instead that player does nothing.
- 5.2.6. Some cards refer to players taking "the same action" or "different actions".
- Actions a player performs are the same if they are all the same basic action or if all of those actions were initiated by the same ability from the same card.
 - Actions a player performs are different if each of them is initiated from a different basic action or card ability. Instances of equivalent abilities on different cards are still different actions.

Example: The Corp, playing MirrorMorph: Endless Iteration, plays Hedge Fund as the first action of their turn, then plays Lateral Growth. Even though two different cards were played, both times the Corp used the basic action to "Play 1 operation from HQ". Since their first two actions were the same, the Corp will not be able to trigger the ability on MirrorMorph this turn.

5.2.7. Corp Actions

- The following BASIC ACTIONS are always available to the Corp during their action phase, in addition to any actions provided by card abilities.
- "[click]: Gain 1[c]."
- "[click]: Draw 1 card."
- "[click]: Install 1 agenda, asset, upgrade, or piece of ice from HQ." See [section 8.5](#).
- "[click]: Play 1 operation from HQ." See [section 8.6](#).
- "[click], 1[c]: Advance 1 installed card." See [section 1.18](#).
- "[click], 2[c]: Trash 1 resource. **Take this action only** if the Runner is tagged." See [section 10.5](#).
- "[click][click][click]: Purge virus counters." See [section 10.1.2](#).

5.2.8. Runner Actions

- The following BASIC ACTIONS are always available to the Runner during their action phase, in addition to any actions provided by card abilities.
- "[click]: Gain 1[c]."
- "[click]: Draw 1 card."

- d. “[click]: Install 1 program, resource, or piece of hardware from the grip.” See [section 8.5](#).
- e. “[click]: Play 1 event from the grip.” See [section 8.6](#).
- f. “[click]: Run any server.” See [section 6](#).
- g. “[click], 2[c]: Remove 1 tag.” See [section 10.5](#).

5.3. Draw Phase

- 5.3.1. The draw phase is the first phase of the Corp’s turn. It includes the formal beginning of the Corp’s turn and the Corp’s mandatory draw.
- 5.3.2. The MANDATORY DRAW is a card draw performed by the Corp. It is not an action and does not cost [click].
- 5.3.3. The Runner’s turn does not have a draw phase.

5.4. Action Phase

- 5.4.1. The action phase is the first phase of the Runner’s turn and the second phase of the Corp’s turn. It is the only timing structure in which players can take actions, as discussed in section 5.2.
- 5.4.2. During a player’s action phase, they must continue to take actions until they have no more clicks or the phase is ended directly by a card ability. The player may take actions in any combination and order as long as they pay the appropriate costs. Once they have no remaining clicks, the action phase ends.
- 5.4.3. A player’s action phase can be forced to end by certain card abilities (e.g. via the ability on a **terminal** operation or event). When a player’s action phase is forced to end, the following occurs:
 - a. The current step of the action phase ends, and all remaining steps are skipped.
 - b. Any currently open priority windows close, their associated abilities lose their pending status, and the player with priority loses priority.
 - c. Any currently resolving actions or abilities cease resolving and are considered complete.
 - d. A checkpoint occurs, following which the game proceeds to the active player’s discard phase.

5.5. Discard Phase

- 5.5.1. The discard phase is the last phase of each player's turn. It requires the active player to discard cards until their maximum hand size is satisfied and includes the formal end of that player's turn.
- 5.5.2. DISCARDING is the act by which a player moves a card to their discard pile during their discard phase if they have exceeded their maximum hand size.
 - a. Cards discarded from HQ are always sent to Archives facedown, regardless of whether they have been previously accessed by the Runner.
 - b. A discarded card is not considered to have been trashed, and vice versa.
- 5.5.3. A player's MAXIMUM HAND SIZE is the maximum number of cards that player can keep in their hand during their discard phase.
 - a. Players begin the game with a maximum hand size of five cards.
 - b. Each point of brain damage the Runner suffers permanently reduces their maximum hand size by one card. See [section 10.4](#). Other abilities can modify a player's maximum hand size directly.

5.6. Steps of the Corp's Turn

- 5.6.1. During the draw phase, the following steps occur in order:
 - a. The Corp gains their allotted clicks (default [click][click][click]).
 - b. A paid ability window occurs, in which players may use paid abilities, the Corp may rez non-ice cards, and the Corp may score agendas. (P) (R) (S)
 - c. The Corp's recurring credits ([recurring]) refill.
 - d. The Corp's turn formally begins. **Conditions related to the turn beginning are met.**
 - e. The Corp performs their mandatory draw.
 - f. The Corp's draw phase is complete. Play proceeds to the Corp's action phase.
- 5.6.2. During the Corp's action phase, the following steps occur in order:
 - a. A paid ability window occurs, in which players may use paid abilities, the Corp may rez non-ice cards, and the Corp may score agendas. (P) (R) (S)
 - b. If the Corp has any unspent [click], the Corp takes an action. Otherwise, skip to (d).
 - c. Return to (a).

d. The Corp's action phase is complete. Play proceeds to the Corp's discard phase.

5.6.3. During the Corp's discard phase, the following steps occur in order:

- a. The Corp discards one card at a time from HQ until the number of cards in HQ is less than or equal to the Corp's maximum hand size.
- b. A paid ability window occurs, in which players may use paid abilities and the Corp may rez non-ice cards. (P) (R)
- c. If the Corp has any unspent [click], the Corp loses those [click].
- d. The Corp's turn formally ends. **Conditions related to the turn or discard phase ending are met.**
- e. The Corp's discard phase and turn are complete. Play proceeds to the Runner's turn.

5.7. Steps of the Runner's Turn

5.7.1. During the Runner's action phase, the following steps occur in order:

- a. The Runner gains allotted clicks (default [click][click][click][click]).
- b. A paid ability window occurs, in which players may use paid abilities and the Corp may rez non-ice cards. (P) (R)
- c. The Runner's recurring credits ([recurring]) refill.
- d. The Runner's turn formally begins. **Conditions related to the turn beginning are met.**
- e. A paid ability window occurs, in which players may use paid abilities and the Corp may rez non-ice cards. (P) (R)
- f. If the Runner has any unspent [click], the Runner takes an action. Otherwise, skip to (h).
- g. Return to (e).
- h. The Runner's action phase is complete. Play proceeds to the Runner's discard phase.

5.7.2. During the Runner's discard phase, the following steps occur in order:

- a. The Runner discards one card at a time from the grip until the number of cards in the grip is less than or equal to the Runner's maximum hand size.
- b. A paid ability window occurs, in which players may use paid abilities and the Corp may rez non-ice cards. (P) (R)

- c. If the Runner has any unspent [click], the Runner loses those [click].
- d. The Runner's turn formally ends. **Conditions related to the turn or discard phase ending are met.**
- e. The Runner's discard phase and turn are complete. Play proceeds to the Corp's turn.

6. Runs

6.1. General

- 6.1.1. A RUN is a Runner's attack on a server. There are 6 phases used in the procedure of carrying out a run, described in sections [6.3](#) through [6.8](#). A run always begins with the Initiation Phase. [Section 6.9](#) contains the full set of steps carried out during each phase, and should be referred to in conjunction with the sections discussing the phases.
- 6.1.2. The ATTACKED SERVER is the server the Runner is attempting to reach, usually to **breach** that server.
 - a. The Runner announces the attacked server in the initiation phase of the run.
 - b. Card abilities can sometimes change the attacked server during the run. This does not end the run, but it can affect whether or not certain card abilities or effects apply.
 - c. **If an ability changes the Runner's position, the attacked server and the current timing step of the run could also change. See [section 6.2.5](#).**
 - d. **A few abilities change the attacked server directly, without referring to the Runner's position. These abilities do not change the current timing step of the run.**

Example: Sneakdoor Beta's ability changes the attacked server from Archives to HQ. The Runner does not need to encounter ice protecting HQ after resolving this ability.

- 6.1.3. Many abilities modify the sequence of steps taken to resolve a run. Such abilities can direct players to skip steps or phases, replace steps with other effects, move directly from the current timing point to a new phase, or make other changes to the procedures explained in this section. Heed the Golden Rules in [section 1.2](#).
 - a. **Some abilities direct the Runner to approach or encounter a piece of ice as part of moving them to that ice's position. See [section 6.2.5](#).**
 - b. **One card, *Code Replicator*, directs the Runner to approach the piece of ice in their position without first moving them to a new position. This effect changes the current timing step of the run to [step 6.9.2](#), the Approach Ice Phase.**

- c. An ability that directs the Runner to encounter a piece of ice **without first moving the Runner's position** does not change the current timing point of the run, **the attacked server, or the Runner's position. The effect of such an ability is to resolve an Encounter Ice Phase independent of the current state of the run.** See [section 6.5.9](#).
- d. Any time **the current timing step of the run is moved** to a different phase of the run, except for an instruction to end the run (See [rule 6.1.4](#), below), no more steps in the previously-active phase are carried out. If a paid ability window is open, players complete that window normally. Then the active phase is complete and a checkpoint occurs. Finally, the game proceeds to the new phase.
- e. **If a condition refers to encountering a piece of ice "after" an approach or passing a piece of ice "after" an encounter, then that condition is only met by the indicated phases or steps occurring in direct sequence according to the standard progression of the run.**

Example: The Runner uses Inversificator to break the subroutine on a Mirāju that is protecting Archives. When the encounter ends, Mirāju's ability creates a replacement effect so that the Runner moves to the outermost position of Archives instead of passing Mirāju. Because the ice is not passed, Inversificator's ability does not meet its trigger condition. Later, the Runner encounters the same Mirāju again and does not break its subroutine. When this encounter ends, the Runner passes Mirāju. Even though the Runner has used Inversificator to fully break Mirāju during an encounter earlier in the run, Inversificator's trigger condition is not met when the Runner passes Mirāju because the Runner is not passing it after the relevant encounter.

- 6.1.4. To END THE RUN is to halt the Runner's progress toward the server. When an effect ends the run, the current phase of the run ends without following any of its remaining steps, **and before completing open priority windows. The game immediately** moves from the current timing point to [step 6.9.6](#), the Run Ends **Phase**.
- 6.1.5. JACKING OUT is the process by which a Runner voluntarily ends a run. Jacking out follows the usual process for ending the run as described in [rule 6.1.4](#), but some card abilities function differently depending on whether or not the Runner chose to end the run by jacking out.
 - a. **The Runner has the opportunity to jack out after passing a piece of ice.**
 - b. **If there is no ice protecting the server, the Runner has the opportunity to jack out before approaching the server.**

6.2. Position

- 6.2.1. The Runner's POSITION during a run is a measure of their distance from the server. The Runner's possible positions correspond to the positions of ice protecting the attacked server.

- a. The Runner does not have a position while they are not engaged in a run, nor do they maintain their position from one run to the next. **After a run initiates**, the Runner **moves to** the outermost position **of the attacked server, if it has any**.
- 6.2.2. The Runner progresses through the positions protecting a server (and corresponding ice) in order. They approach, encounter (if the ice is rezzed), and pass each piece of ice, then proceed to the next piece of ice moving inward, until they pass the innermost piece of ice or the run is ended.
- a. **Once the Runner leaves the position of the innermost piece of ice protecting a server, they no longer have a position. This occurs at [step 6.9.4d](#) of the Movement Phase. However, abilities can still move the Runner to a position later in that phase.**
 - b. **During the Success Phase and the Run Ends Phase, the Runner has no position and can no longer move to a position.**

*Example: The Corp uses **Ganked!** to force the Runner to encounter Cell Portal in the middle of breaching a server. Since the run is already successful, resolving Cell Portal's subroutine will have no meaningful effect except to **derez Cell Portal**.*

- 6.2.3. Ice entering or leaving a position protecting the attacked server can affect the Runner's position.
- a. If a piece of ice is added to or removed from the attacked server in a position outward from the Runner's current position, nothing happens. The Runner will normally not need to approach or encounter the new piece of ice to reach the server.
- Example: During a run on R&D, **Architect's subroutines resolve** and the Corp installs a new piece of ice in the outermost position protecting this server. The Runner does not approach that new piece of ice during this run because they have already passed that position.*
- b. If a piece of ice is added to the attacked server in a position inward from the Runner's current position, the Runner and each piece of ice outside of the new ice's position shift one space outward.
- Example: During a run on a remote server, **Howler's subroutine resolves**, and the Corp installs a new piece of ice in the next inward position protecting the server. The Runner will have to approach that new piece of ice later in this run because they have not passed that position.*
- c. If a piece of ice inward from the Runner's position is uninstalled or moved to another server, the Runner and each piece of ice outside of the departed ice's position shift one space inward. The Runner does not re-approach ice they have already passed.

6.2.4. Changes to the piece of ice in the Runner's current position can affect the progression of the run, but they do so differently during the Approach Ice Phase than during the Encounter Ice Phase.

- a. If a piece of ice is uninstalled or moved to another position (protecting this or another server) while it is being approached, the **run** moves to the **Movement** Phase, all outward ice moves one position inward, and the run continues. The Runner does not re-approach ice they have already passed.

Example: The Runner has passed 1 piece of ice and is approaching the innermost piece of ice, which is a rezzed Himitsu-Bako. During the paid ability window of the approach, the Corp uses the ability on Himitsu-Bako to add it to HQ. The Runner passes this position and will approach the server next, while the other piece of ice protecting the server moves to the innermost position.

- b. If a piece of ice is uninstalled or derezzed while it is being encountered, the **run** moves to the **Movement** Phase. All outward ice moves one position inward if the ice was uninstalled, and the run continues. The Runner does not re-approach ice they have already passed.
- c. If a piece of ice currently being encountered moves to another position or is swapped with installed ice in another position, the Runner remains with the encountered ice and continues the run from that ice's new position.

Example: The Runner encounters Bullfrog as the innermost piece of ice protecting a remote server. Bullfrog's subroutine resolves and it is moved to be the outermost piece of ice protecting Archives. The Runner is now running on Archives and continues the run from Bullfrog's new position.

- d. If a piece of ice currently being approached is swapped with ice in another position, the Runner remains in their current position and is now approaching the new piece of ice occupying that position.
- e. If a piece of ice is uninstalled, derezzed, moved, or swapped **after it** is passed, the Runner is not moved.

6.2.5. During a run, abilities can move the Runner directly to a position.

- a. **If an ability instructs the Runner to move to a piece of ice in a different position, the Runner's position becomes the position corresponding to that ice, and the server that ice protects becomes the attacked server. The current timing step becomes [step 6.9.2](#), the Approach Ice Phase, or [step 6.9.3](#), the Encounter Ice Phase, according to whether the Runner was directed to approach or encounter the ice.**
- b. **If an ability instructs the Runner to move to "the outermost position" of a server, that server becomes the attacked server. If there is any ice protecting that server, the Runner's position becomes the position of the outermost such piece of ice, and the current timing step becomes [step 6.9.2](#), the**

Approach Ice Phase. If there is no ice, the Runner ceases to have a position, and the current timing step becomes [step 6.9.4](#), the Movement Phase.

- c. If an ability instructs the Runner to move to a position during the Success Phase, during the Run Ends Phase, or while there is no run in progress, instead the Runner does nothing. See [rule 6.2.2b](#).
- d. If an ability instructs the Runner to move to and approach a position they are already approaching, instead the Runner does nothing.

6.3. Initiation

6.3.1. The INITIATION PHASE **sets the parameters for the run and begins the run.**

6.3.2. **At the start of** the Initiation Phase, the Runner announces a server to be the attacked server.

- a. Some abilities state that the Runner “cannot run on” or “cannot initiate a run on” certain servers under certain conditions. These effects refer to the announcement of the attacked server in the Initiation Phase.

Example: Off the Grid states in part, “The Runner cannot initiate a run on this server.” This only prohibits the Runner from announcing this particular remote server as the attacked server at the beginning of the run. If an ability changes the attacked server while a run is in progress, Off the Grid does not interfere with that ability.

- b. Some abilities impose an additional cost to run, either generally or on a particular server. Any costs to run a server must be paid at the time that server is announced as the attacked server.

6.3.3. **After the attacked server is announced**, the Runner receives credits equal to the number of bad publicity the Corp has. See [section 10.6](#) for an explanation of bad publicity.

- a. Credits from bad publicity cannot be spent to pay costs to make the run, as they are received after the attacked server is announced.
- b. Once they are received, credits from bad publicity are in the Runner’s credit pool and behave just as any other credits. They can be spent, lost, placed on cards, and so on as normally allowed.
- c. During the Run Ends Phase, any credits from bad publicity remaining in the Runner’s credit pool are lost.

6.3.4. Some abilities look for particular conditions occurring “during a run” or are restricted to being used “during a run.” For purposes of abilities, the run formally begins only after the attacked server is announced and any costs are paid.

Example: The Runner initiates a run against a server with a rezzed Heinlein Grid while the Corp has Enhanced Login Protocol in play. They do not lose any credits to Heinlein Grid’s ability, as

the additional click Enhanced Login Protocol requires them to spend is spent to initiate the run, and is not spent during the run.

- 6.3.5.** When the Initiation Phase completes, the run moves either to the Approach Ice Phase ([section 6.4](#)) or the Movement Phase ([section 6.6](#)). If there is ice protecting the attacked server, the Runner's position becomes the outermost piece of ice, and the run proceeds to the Approach Ice Phase. If there is no ice protecting the server, then the run moves directly to the Movement Phase.

6.4. Approach Ice

- 6.4.1.** The APPROACH ICE PHASE gives the Corp the opportunity to rez the ice in the Runner's current position.
- 6.4.2.** For purposes of meeting trigger conditions, the Runner approaching a piece of ice is equivalent to the Approach Ice Phase beginning at that ice's position. Abilities with this trigger condition are subject to [rule 9.2.8f](#).
- 6.4.3.** The paid ability window in [step 6.9.2b](#) of the Approach Ice Phase is the only time the Corp can normally rez ice. In this window, the Corp only gains the ability to rez the piece of ice being approached, not any other ice.
- 6.4.4.** When the Approach Ice Phase completes, the run moves either to the Encounter Ice Phase ([section 6.5](#)) or the Movement Phase ([section 6.6](#)). If the approached piece of ice is rezzed, the Runner encounters it next. If the approached piece of ice is unrezzed or if there is no longer ice installed in that position, then the run moves directly to the Movement Phase.

6.5. Encounter Ice

- 6.5.1.** The ENCOUNTER ICE PHASE gives the Runner the opportunity to interact with the ice in their current position, then requires the Corp to resolve unbroken subroutines on that ice.
- 6.5.2.** For purposes of meeting trigger conditions, the Runner encountering a piece of ice is equivalent to the Encounter Ice Phase beginning at that ice's position. Abilities with this trigger condition are subject to [rule 9.2.8f](#).
- 6.5.3.** As each encounter begins, each subroutine on the encountered ice gains an "unbroken" status associated with that encounter. See [section 9.8](#) for details.
- 6.5.4.** During the paid ability window in the Encounter Ice Phase ([step 6.9.3b](#)), the Runner can break subroutines on the ice. See [rule 9.8.6](#).
- 6.5.5.** After the paid ability window in the Encounter Ice Phase closes, unbroken subroutines on the encountered ice resolve. See [section 9.8.8](#).

6.5.6. At the end of the Encounter Ice Phase, the run continues to the Movement Phase, [section 6.6](#).

6.5.7. Fully Break

- a. During each encounter, the Runner FULLY BREAKS the encountered ice the first time all subroutines on that ice are broken.
- b. Whenever the Runner fully breaks a piece of ice, if all its subroutines were broken using abilities on a single object, that object also FULLY BREAKS the ice.
- c. If an encountered piece of ice has no subroutines, the Runner fully breaks it when [step 6.9.3b](#) of the encounter begins. No objects fully break the ice in this case.
- d. If the encountered piece of ice gains subroutines after the Runner fully breaks it, the Runner's status of having fully broken the ice remains in place. Conditions that check whether the Runner fully broke the ice will still be met as normal. Breaking the new subroutines will not fully break the ice or meet related conditions again.

6.5.8. Bypass

- a. When an effect allows the Runner to BYPASS a piece of ice, the Encounter Ice Phase is aborted and the Runner immediately proceeds to pass that ice.
- b. Most bypass effects occur at the beginning of an encounter and **result in** steps [6.9.3b](#) and [6.9.3c](#) **never occurring**, meaning that subroutines on the encountered ice are neither broken nor resolved.
- c. A few bypass effects can be triggered later. If ice is bypassed in [step 6.9.3b](#), subroutines may or may not have been broken, but [step 6.9.3c](#) **never occurs** so no subroutines resolve. Note that for ice with zero subroutines, the Runner has "broken all subroutines" if and only if [step 6.9.3b](#) was reached. See [rule 9.12.2c](#).

Example: The Runner plays Forked and encounters a Troll that they can bypass with Femme Fatale. The Runner pays 0[c] to bypass the Troll with Femme Fatale. Forked does not meet its trigger condition, and Troll is not trashed, because the opportunity to break subroutines was never reached.

6.5.9. Forced Encounters

- a. If an ability directs the Runner to encounter a piece of ice **without first changing their position**, resolve an Encounter Ice Phase but do not change the Runner's position, if any. After the encounter, return to the effect that caused the encounter and proceed from there. This type of encounter outside the normal progression of a run is referred to as a FORCED ENCOUNTER.

Example: If Shiro causes Chrysalis to be accessed, resolve the encounter with Chrysalis and then return to resolving subroutines on Shiro.

- b. If a forced encounter occurs during a run, an instruction to end the run during that encounter is applied to both the Encounter Ice Phase being resolved and the phase from which the additional encounter was initiated. If a forced encounter occurs outside of a run, an instruction to end the run simply **closes** the Encounter Ice Phase and returns the game to the effect that called for the encounter.

*Example: **The Corp uses The Twins to force** the Runner to encounter a piece of ice again **when they pass it**. If a subroutine directs the Corp to end the run during the extra encounter, both that encounter and the original **Movement Phase** are aborted, and the game proceeds to the Run Ends phase.*

- c. **An instruction that creates a forced encounter is not finished resolving until the encounter is complete.**
- d. **During a forced encounter, if an ability moves the Runner or directs the Runner to approach a piece of ice, that movement resolves and changes the parameters of the underlying run normally, but does not end or interfere with the forced encounter.**

6.6. Movement

- 6.6.1. **The MOVEMENT PHASE handles the progression of the run between pieces of ice and after the innermost piece of ice, up to the point when the Runner approaches the attacked server.**
- 6.6.2. **If the Movement Phase was reached from the Approach Ice Phase or the Encounter Ice Phase, the Runner passes the ice in their position.**
- 6.6.3. **During each Movement Phase, the Runner has the opportunity to jack out. See [section 6.1.5](#).**
- 6.6.4. **If the Runner declines to jack out, their position moves to the next inward position if there is one.**
- 6.6.5. **If the Runner moves to a new position, the run continues from the Movement Phase to the Approach Ice Phase, [section 6.4](#). The Runner will approach the piece of ice in their new position.**
- 6.6.6. **If the Runner does not jack out or return to the Movement Phase, they approach the server. This has no inherent effects, but is referred to by some card abilities. After the Runner approaches the server, the run continues to the Success Phase, [section 6.7](#).**

6.7. Success

- 6.7.1. **The SUCCESS PHASE declares the run successful and allows the Runner to breach the attacked server.**
- 6.7.2. A run during which the runner breaks through the Corp's defenses and reaches the server itself is declared a SUCCESSFUL RUN. This label is frequently referred to by card abilities.
- 6.7.3. **Declaring a run to be successful is not the same as the Success Phase beginning. Accordingly, abilities that have trigger conditions related to a successful run are not subject to [rule 9.2.8f](#).**
- 6.7.4. Many abilities that initiate a run contain a conditional ability with the trigger condition "If successful". This means, "After the run created this way becomes successful".
- If an effect that initiates a run specifies that that run must be made against a specific server or servers, any "If successful" abilities associated with that effect are tied to that server or servers. If the attacked server has changed since the run was initiated, and no longer is a server that could have been chosen for this run, the "If successful" abilities do not meet their trigger conditions.

Example: The Runner plays Because I Can, which directs them to make a run on a remote server. If an effect moves the run to a central server, the "If successful" effect on Because I Can will not apply. However, if the run is moved instead to another remote server, the "If successful" effect is still applied, since Because I Can would have allowed a run on the second remote server to begin with.

- "If successful" abilities frequently create lingering effects (see [section 9.10](#)) that modify the procedure for accessing cards or replace **breaching** the attacked server with another effect entirely. These effects apply to the **breach** permitted by the rules for this run (see [section 6.7.5](#)), and they expire when the run is complete.
- If an "If successful" effect gives the Runner an effect they can optionally carry out instead of **breaching**, that decision is made at the time **the breach** would begin, in [step 6.9.5b](#).

Example: The Runner plays Account Siphon while Ash 2X3ZB9CY is installed in the root of HQ, and the run is successful. The Runner can wait until after the trace from Ash 2X3ZB9CY completes before deciding to access cards normally or force the Corp to lose credits.

- 6.7.5. When the Runner **breaches the server**, they follow the procedures detailed in [section 7](#).

6.8. Run Ends

6.8.1. The RUN ENDS PHASE formalizes the end of the run and prepares the game to return to the timing structure from which the run was initiated.

6.8.2. At the beginning of the Run Ends Phase, the game processes each priority window that was open when the run was ended. This is done according to the following procedure:

- a. If a paid ability window was open when the run was ended, the window closes. No player can trigger further paid abilities or resolve other options the window would normally allow (such as rezzing cards).

Example: The Corp spends a counter from a scored Nisei Mk II to end the run. Once this happens, the Runner does not have an opportunity to use their bad publicity credits from this run to pay for using Self-Modifying Code.

- b. If the run was ended during a reaction window opened due to a phase beginning, the reaction window closes, as per [rule 9.2.8f](#). No other pending abilities are resolved.

Example: The Runner uses the ability on Security Nexus during a reaction window at the beginning of the Encounter Ice Phase. The trace is successful, so the ability ends the run. If Tollbooth's ability is also pending in this reaction window, it will not be resolved, since the encounter has ended.

- c. Any other open priority window is completed normally. If more than one priority window needs to be completed this way, they are resolved in the usual order described in [rule 9.1.2a](#).

*Example: The Corp resolves the ability on Giordano Memorial Field during a reaction window in the **Success Phase**. This ability ends the run, but first the priority window is completed. If Embolus's second ability is pending in the same reaction window (as it usually will be if both cards are rezzed), the Corp must remove a power counter from Embolus.*

6.8.3. During the Run Ends Phase, if the Runner has any credits from bad publicity remaining in their credit pool, the Runner loses those credits.

6.8.4. As part of the Run Ends Phase, the game determines whether the run should be declared an UNSUCCESSFUL RUN. This is a label applied to runs in which the Runner failed to reach the server they attacked, and is referred to by card abilities. The run is declared unsuccessful if neither of the following conditions have been met:

- a. If the run reached [step 6.9.5](#), the **Success Phase**, it is not declared unsuccessful. This is the case regardless of whether the run was actually declared successful.

Example: The Runner reaches a server with Crisium Grid rezzed. Crisium Grid's ability stops the run from being declared successful, but the run is not declared unsuccessful either.

- b. If the attacked server has ceased to exist, the run is not declared unsuccessful. See [rule 8.5.8](#).

- 6.8.5. Conditions related to a run ending are not met as soon as an “end the run” effect resolves or at the beginning of the Run Ends Phase. Instead, these conditions are met at [step 6.9.6d](#), when the entire run timing structure is complete. Similarly, static abilities that apply during a run and durations that apply until the end of a run expire at [step 6.9.6d](#).**

Example: The Runner plays The Noble Path to make a run in which they prevent all damage. During the run, the subroutine on Chum resolves, and then the Runner encounters Wall of Static and does not break its subroutine. When the “end the run” ability resolves, the encounter also ends, so the trigger condition on Chum's delayed conditional ability is met. The ability will resolve and fail to damage the Runner during the Run Ends Phase, because The Noble Path's effect still applies. Similarly, Georgia Emelyov's first ability meets its trigger condition at [step 6.9.6c](#), which is still before the run is over, so the damage will be prevented.

Example: The Runner plays The Noble Path to make a run in which they prevent all damage. The run is successful but the Runner takes a tag during the run. When the run ends, Dedicated Response Team's ability meets its trigger condition at the same time that The Noble Path's effect expires, so the Runner will suffer 2 meat damage.

6.9. Steps of a Run

6.9.1. Initiation Phase

- a. The Runner announces the attacked server.
- b. The Runner gains 1[c] to spend during the run for each bad publicity the Corp has.
- c. The run formally begins. **Conditions related to the run beginning or initiating are met.**
- d. The Initiation Phase is complete. If the attacked server has at least one piece of ice protecting it, **proceed to [step 6.9.2](#), Approach Ice Phase**, approaching the outermost piece of ice protecting the attacked server. If the attacked server is not protected by ice, **proceed to [step 6.9.4](#), Movement Phase**.

6.9.2. Approach Ice Phase

- a. **The approach begins. Conditions related to the Runner approaching this ice are met.**

- b. A paid ability window occurs, in which players may use paid abilities and the Corp may rez non-ice cards. Additionally, the Corp **can** rez the approached piece of ice. (P) (R)
- c. The Approach Ice Phase is complete. If the approached piece of ice is rezzed, continue to [step 6.9.3](#), Encounter Ice **Phase**. **Otherwise, proceed** to [step 6.9.4](#), **Movement Phase**.

6.9.3. Encounter Ice Phase

- a. **The encounter begins. Conditions related to the Runner encountering this ice are met.**
- b. A paid ability window occurs, in which players may only use paid abilities. Subroutines on the encountered ice can be broken during this window. (P)
- c. **If there are unbroken subroutines that have not yet resolved this encounter, the Corp resolves the next such subroutine in order. Otherwise, skip to (e).**
- d. **Return to (c).**
- e. The Encounter **Ice** Phase is complete. Continue to [step 6.9.4](#), **Movement Phase**.

6.9.4. Movement Phase

- a. **If the Runner's position corresponds to a piece of ice, the Runner passes that ice. Conditions related to passing ice are met. If there are no positions more inward than the ice just passed, conditions related to passing all of the ice protecting a server are also met.**
- b. **A paid ability window occurs, in which players may only use paid abilities. (P)**
- c. **The Runner may choose to jack out, proceeding to [step 6.9.6](#), Run Ends Phase. If the Runner does not, continue to (d).**
- d. **The Runner's position moves to the next position inward, if any.**
- e. **A paid ability window occurs, in which players may use paid abilities and the Corp may rez non-ice cards. (P) (R)**
- f. **If the Runner moved to a new position, return to [step 6.9.2](#), Approach Ice Phase, approaching the piece of ice in their new position. If not, continue to (g).**
- g. **The Runner approaches the attacked server. Conditions related to the Runner approaching a server are met.**
- h. **The Movement Phase is complete. Continue to [step 6.9.5](#), Success Phase.**

6.9.5. Success Phase

- a. The run is declared successful. **Conditions related to a successful run are met.**
- b. The Runner **breaches** the attacked server.
- c. The **Success Phase** is complete. Continue to [step 6.9.6](#), Run Ends **Phase**.

6.9.6. Run Ends Phase

- a. Any priority windows that were open when the run moved to this phase are completed or closed as described in [section 6.8.2](#).
- b. The Runner loses any unspent bad publicity credits.
- c. If **the Success Phase** was not reached during this run and the attacked server still exists, the run becomes unsuccessful.
- d. The Run Ends Phase and the run as a whole are both complete.

7. Accessing Cards **and Breaching Servers**

7.1. Accessing Cards

- 7.1.1. **ACCESSING** is the process of the Corp showing one of their cards to the Runner for potential interaction.
- 7.1.2. While the Runner is accessing a card, the Runner is allowed to look at that card, even if it would normally not be visible to them. While the Runner is accessing a card, the Corp is allowed to look at it as well, except when the card is being accessed from R&D.
- 7.1.3. An ability that meets its trigger condition when its source card is accessed is active even if the source card is inactive. See [section 9.1.8](#).
- 7.1.4. During each access, the Runner has 1 opportunity to use a mid-access ability. This occurs after the reaction window at the beginning of the access, and before the Runner steals an accessed agenda.
- 7.1.5. The Runner always has the ability "Access → Pay the trash cost of the accessed card: Trash it." This is the BASIC TRASH ABILITY.
 - a. All assets and upgrades have trash costs, as do some ice and operations. If a card does not have a trash cost, the Runner cannot pay its trash cost, and therefore the Runner cannot use the basic trash ability during that access.

- b. The Runner cannot trash or pay the trash cost of a card in the Corp's discard pile, either with the basic trash ability or with other mid-access abilities.
 - c. One card, *Mumbad Virtual Tour*, has the ability "If the Runner accesses this upgrade while it is installed, they must trash it, if able." When this ability applies, the Runner can use any mid-access ability they have available that trashes the card, but they cannot decline to use a mid-access ability or use one that would not trash the accessed card unless they are unable to pay the cost of any ability that would trash it.
 - d. One card, *Neutralize All Threats*, has the ability "The first time each turn you access a card with a trash cost, reveal it. You must trash that card by paying its trash cost, if able." When this ability applies, the Runner can only use the basic trash ability. They cannot decline to use a mid-access ability or use a mid-access ability available from a card unless they cannot pay the cost of the basic trash ability.
- 7.1.6. If, after resolving mid-access abilities, the Runner is still accessing an agenda, they must steal it.
- a. Some abilities can create additional costs to steal an agenda. The Runner can decline to pay such costs and not steal the agenda. See [section 1.16.9](#).
- 7.1.7. If a card moves to another location while it is being accessed, the access ends immediately.
- 7.1.8. If a run is in progress, an instruction to end the run ends any access taking place. If an access is occurring outside of a run, an instruction to end the run has no effect on that access.
- 7.1.9. One card, *Information Sifting*, uses a method other than a breach to establish a set of cards for the Runner to access. The Runner accesses those cards one at a time in the order of their choice. Each access is performed as a separate instruction. See [rule 9.11.4b](#). Accessing cards in this way is an exception to [rule 9.12.2a](#).
- 7.1.10. One card, *Counter Surveillance*, instructs the Runner to access a certain number of cards from a server without mentioning a breach. The procedure for these accesses follows the normal rules for breaching that server except: there is no random access limit if the server is HQ or R&D; and the procedure ends once the designated number of cards have been chosen for access.

7.2. Steps of Accessing a Card

- 7.2.1. The card is accessed. Conditions related to accessing this card are met.
- 7.2.2. The Runner may use a single mid-access ability, such as the basic trash ability.
- 7.2.3. If the accessed card is an agenda, the Runner must steal it.

7.2.4. The access is complete. Conditions related to an access ending are met.

7.3. Breaching Servers

- 7.3.1. BREACHING a server is the process of the Runner accessing a particular set of cards associated with that server. Most accesses occur during a breach. [Step 6.9.5b](#) of a run calls for the Runner to breach the attacked server, but card abilities can also directly instruct the Runner to breach a server.
- 7.3.2. When the Runner breaches Archives, all the facedown cards in the Corp's discard pile are turned faceup before the Runner accesses any cards.
- If a card is added to Archives facedown after this step, that card remains facedown for the remainder of the breach, even after it is accessed. The Runner can only look at a facedown card in Archives while accessing it.
- 7.3.3. The cards associated with a breached server that the Runner could potentially access are called CANDIDATES. The set of candidates is compiled from the cards in the root of the breached server and, for central servers, the cards in the zone that corresponds to that server (see [rule 4.6.7a](#)). [Section 7.4](#) defines which of these cards are candidates.
- 7.3.4. During a breach, the Runner is presented with the current candidates. They choose 1 of those cards and access it. This process repeats until there are no longer any candidates.
- When presented with candidates during a breach of HQ, the Runner chooses either a specific candidate that is not in the Corp's hand or to access a random candidate from among the ones in the Corp's hand. A card randomly chosen this way is treated as though the Runner had chosen it.
 - One card, *Dedicated Neural Net*, instructs the Corp to choose which cards the Runner accesses from HQ. While this effect applies, the Runner still chooses candidates that are not in the Corp's hand as normal. Whenever the Runner chooses to access a candidate from the Corp's hand, instead of a candidate being chosen at random, the Corp chooses 1 of those candidates for the Runner to access. The chosen card is still treated as though the Runner had chosen it.
- 7.3.5. The Runner can only choose candidates from the Corp's hand or deck a limited number of times during a breach of HQ or R&D, respectively. This number is called the RANDOM ACCESS LIMIT. Before the Runner begins accessing cards during a breach of HQ or R&D, the random access limit for that breach is determined, after which it will not change for the remainder of that breach.
- By default, the random access limit is 1.

- b. If an ability allows the Runner to "access additional cards" during a breach of HQ or R&D, the effect of that ability is to increase the random access limit. Such an ability can only be applied at the beginning of the breach, before the value of the random access limit is set.
- c. Cards are counted against the random access limit in the manner described in [rule 7.3.6a](#).

7.3.6. Some rules and abilities refer to the number of cards the Runner accesses or the number of times the Runner accesses a card.

- a. If such a reference applies while a breach is in progress, it counts the number of times a candidate has been chosen so far during that breach, regardless of how many of those candidates have actually been accessed.

Example: The Runner plays Immolation Script, and the subroutine on Hudson 1.0 resolves before Archives is breached. The Runner will only be able to choose 1 candidate during the breach, regardless of whether they choose to access that card or apply Immolation Script's replacement effect.

- b. If such a reference applies after a breach is complete, it counts only the number of times the Runner actually resolved an access during that breach.

Example: The Runner plays Rip Deal while they have Obelus installed. If they use Rip Deal's ability to replace accesses of cards in HQ with adding cards from the heap to the grip, the cards those replacement effects applied to were not accessed. When Obelus's ability resolves after the breach ends, it will only cause the Runner to draw cards for upgrades they accessed in the root of HQ.

7.3.7. If a run is in progress, an instruction to end the run ends any breach taking place. If there is no run in progress, an instruction to end the run has no effect on a breach.

7.3.8. If a server would be breached while a breach is already in progress, instead the new breach takes place when the current breach ends. The effect creating the delayed breach is treated as a conditional ability controlled by the Runner.

Example: The Runner steals Clone Retirement during a breach of HQ, causing the Corp to take bad publicity. The Runner must complete the current breach before beginning the breach granted from Raymond Flint's first ability.

7.4. Determining Candidates

7.4.1. The candidates at the beginning of a breach are as follows:

- a. Each card in the root of the breached server is a candidate.
- b. If the breached server is HQ, each card in the Corp's hand is a candidate.

- c. If the breached server is R&D, the top card of the Corp's deck is a candidate.
 - d. If the breached server is Archives, each card in the Corp's discard pile is a candidate.
- 7.4.2. If at any time the Runner is prohibited from accessing an object, that object ceases to be a candidate and cannot become a candidate.

Example: The Corp performs a successful trace with the ability on Ash 2X3ZB9CY, prohibiting the Runner from accessing any card other than Ash for the remainder of the current run. When the server is breached, Ash will be the only candidate for the Runner's first selection, and then there will be no candidates remaining, so the breach will end.

- a. One card, *Hudson 1.0*, creates an effect that prohibits the Runner from accessing more than 1 card during a run. This effect always considers whether the Runner has actually accessed a card during that run, even when [rule 7.3.6a](#) would normally apply. If the Runner has accessed any cards, this effect prohibits all accesses, and therefore there will be no candidates.
- 7.4.3. Once the Runner chooses a candidate for access, it ceases to be a candidate for the remainder of that breach. It does not matter whether the chosen candidate is actually accessed. As long as the card is chosen, it ceases to be a candidate.

Example: The Runner is breaching Archives during a run made with Immolation Script. They choose a piece of ice from the set of candidates, and apply Immolation Script's replacement effect to trash another card instead of accessing the chosen ice. After resolving this, the chosen ice is no longer a candidate and cannot be accessed again this breach. (The newly-trashed card, however, could be a candidate.)

Example: The Corp is playing Gagarin Deep Space: Expanding the Horizon, and the Runner breaches a remote server. When the Runner chooses a candidate from the root of that server, an additional cost to access is imposed. The Runner declines to pay the cost and therefore does not perform the access. That card still ceases to be a candidate.

- 7.4.4. When the random access limit is reached during a breach of HQ or R&D, all candidates in the zone corresponding to the breached server cease to be candidates, and no cards in that zone can become candidates for the remainder of that breach. See [section 7.3.5](#).
- 7.4.5. If a candidate leaves the breached server, it ceases to be a candidate. Note that the moved card could be a new object, which could be eligible to become a new candidate if appropriate.

Example: The Runner trashes an installed upgrade during a breach of Archives. That card moves to the Corp's discard pile. The new object for the upgrade's existence in the Corp's discard pile becomes a candidate, so the Runner will access the same physical card twice during this breach.

- 7.4.6. Cards entering a server while a breach is in progress can become candidates.**
- a. If a card enters the root of a server during a breach of that server, it becomes a candidate.
 - b. If a card enters the Corp's hand during a breach of HQ while the random access limit has not yet been reached, that card becomes a candidate.
 - c. If a card enters the Corp's deck during a breach of R&D, that card could become a candidate (either at that time or later in the breach), depending on the random access limit and its position within the deck. See [section 7.4.7](#).
 - d. If a card enters the Corp's discard pile during a breach of Archives, that card becomes a candidate. See also [rule 7.3.2a](#).
- 7.4.7. During a breach of R&D, the Runner is presented with 1 candidate from the Corp's deck at a time in turn, working down from the top of the deck. Cards are made candidates 1 at a time until the access limit is reached.**
- a. After the Runner chooses a candidate from the Corp's deck and whenever any cards enter, leave, or are reordered within the Corp's deck, all cards in the Corp's deck cease to be candidates. Then, if the random access limit has not been reached, the topmost eligible card in the deck becomes a candidate. A card is eligible for this purpose if the Runner has not already chosen to access that object and is not prohibited from accessing that object in this breach.

Example: The Runner plays The Maker's Eye and breaches R&D, accessing 2 additional cards. They access Celebrity Gift and then access and steal Bacterial Programming. The Corp uses Bacterial Programming's ability to rearrange cards in R&D, but they leave Celebrity Gift as the top card of R&D. The cards returned to the top of R&D are new objects, so the Runner must continue the breach from the top of R&D and will access Celebrity Gift again.

Example: The Corp is playing Seidr Laboratories and has Strongbox rezzed in the root of R&D. The Runner breaches R&D with a random access limit of 4. They first access the top card of the Corp's deck and leave it in place. The 2nd card from the top now becomes a candidate, and turns out to be an agenda, which the Runner steals by spending [click] to pay the additional cost imposed by Strongbox. The Corp then uses Seidr Laboratories to add 1 card from Archives to the top of R&D. The 3rd candidate presented to the Runner is the card newly added to the top of R&D. After accessing that card, the next card down is still the object that was accessed first, so the card below that (currently 3rd from the top) becomes the 4th and final candidate.

- b. One card, *Showing Off*, allows the Runner to access cards from the bottom of R&D. While this effect applies, candidates in the Corp's deck are presented

working up from the bottom: treat [rule 7.4.1c](#) as if it said “bottom” instead of “top” and [rule 7.4.7a](#) as if it said “bottommost” instead of “topmost”.

7.5. Steps of Breaching a Server

- 7.5.1. Breaching the server formally begins. Conditions related to breaching this server are met.
- 7.5.2. If the breached server is Archives, turn all cards in the Corp's discard pile faceup.
- 7.5.3. If the breached server is HQ or R&D, determine the limit to how many times the Runner can choose a candidate from the Corp's hand or deck.
- 7.5.4. The Runner chooses a candidate. If they cannot, skip to [step 7.5.7](#).
- 7.5.5. The Runner accesses the chosen card.
- 7.5.6. Return to [step 7.5.4](#).
- 7.5.7. The breach is complete. Conditions related to the breach ending are met.

8. Card Manipulation

8.1. Faceup and Facedown Status

- 8.1.1. An agenda, asset, upgrade, or piece of ice is UNREZZED if it is installed and facedown. An asset, upgrade, or piece of ice is REZZED if it is installed and faceup. Runner cards, faceup agendas, operations, events, and counters are neither rezzed nor unrezzed.
- 8.1.2. To REZ an unrezzed card is to turn it faceup.
 - a. Some paid ability windows allow the Corp to rez assets and upgrades. See [section 9.2.8](#). The paid ability window at [step 6.9.2b](#) of a run allows the Corp to rez ice.
 - b. Card abilities can direct or allow the Corp to rez cards at other times.
 - c. Agendas cannot be rezzed.
 - d. Before rezzing a card, the Corp must pay that card's rez cost. This includes both rezzing in a paid ability window and rezzing through a card ability. Card abilities can modify or ignore rez costs, but must state this explicitly.
 - e. To resolve rezzing a card, the Corp pays its rez cost (accounting for any modifiers or additional costs), a checkpoint occurs (as required by [rule 10.3.4](#)), then the Corp turns the card faceup. For purposes of meeting trigger conditions, the card is considered to be rezzed at the conclusion of this process.

- 8.1.3.** To DEREZ a rezzed card is to turn it facedown.
- a.** Cards are only derezzed by card effects.
 - b.** **There is no inherent cost associated with derezzing a card.**
 - c.** **Derezzing a card is instantaneous and has no component steps.**
- 8.1.4.** Some abilities can install Runner cards facedown or turn already-installed Runner cards facedown.
- a.** Runner cards that are installed or turned facedown do not have a name, type, or subtypes, and their abilities are not active.
 - b.** The Runner can look at their facedown cards at any time.
 - c.** After a facedown Runner card enters the heap, it is turned faceup.
 - d.** A Runner card turned facedown is not considered to be uninstalled and simply remains in the play area.
 - e.** If an installed facedown Runner card is turned faceup, the card gains its name, type, and subtypes, and it becomes active. Turning an installed facedown Runner card faceup does not meet the trigger conditions of “when installed” abilities.
 - f.** **Runner cards are never considered rezzed or unrezzed.**

8.2. Card Movements

- 8.2.1.** **A MOVEMENT is a way that cards change their zone or location following special rules. The movements are: arrange, draw, forfeit, install, play, search, score, steal, swap, and trash.**
- a.** **Other terms that change the locations of cards are not movements. If an ability directs a player to “add” or “move” cards to 1 or more locations, simply put those cards in the indicated locations. If an ability or game rule directs a player to “discard”, “set aside”, or “remove from the game” 1 or more cards, simply put those cards in their owner’s discard pile, the set-aside zone, or the removed-from-game zone, respectively.**
- 8.2.2.** **If a replacement effect modifies the effects of a movement, but does not replace the entire movement by name, the modified effect is still an occurrence of that movement and can still meet trigger conditions relating to that type of movement.**

Example: Harbinger’s ability modifies the effects of trashing it by replacing adding Harbinger to the heap with turning it facedown. Harbinger is still trashed when the modified effect resolves. If this is the first time this turn that the Runner trashed one of their installed cards, they will be able to resolve Wasteland’s ability.

- a. If a movement is replaced or prevented entirely or otherwise cannot be carried out, that movement does not take place and conditions related to that type of movement are not met.

Example: The Corp is resolving the first subroutine on Rototurret to trash an installed program. The Runner uses Sacrificial Construct to prevent the program from being trashed. Since trashing did not occur, the Runner does not place a power counter on District 99, and District 99 can still meet its trigger condition if a program or piece of hardware is successfully trashed later in the turn.

- 8.2.3. To ARRANGE a set of cards is to reposition them among their current locations. The full rules for arranging are detailed in [section 8.3](#).
- 8.2.4. To DRAW a given number of cards is to move that many cards from the top of a player's deck to their hand. The full rules for drawing cards are detailed in [section 8.4](#).
- 8.2.5. To FORFEIT an agenda is to move it from a player's score area to the removed-from-game zone. When a player forfeits an agenda, its agenda points no longer contribute to that player's score, and any objects hosted on it are trashed.
- 8.2.6. To INSTALL a card is to move it from its current zone to the play area. The full rules for installing cards are detailed in [section 8.5](#).
- 8.2.7. To PLAY an event or operation is to move it from its current zone to the play area, resolve its play abilities, and then trash it. The full rules for playing events and operations are detailed in [section 8.6](#).
- 8.2.8. To SEARCH is to look at a specified zone and potentially set aside cards from that zone. The full rules for searching are detailed in [section 8.7](#).
- 8.2.9. To SCORE an agenda is to move it from its current zone to the Corp's score area. The full rules for scoring are detailed in [section 1.17](#).
- 8.2.10. To STEAL an agenda is to move it from its current zone to the Runner's score area. The full rules for stealing are detailed in [section 1.17](#).
- 8.2.11. To SWAP a pair of cards is to move each card to the other's location simultaneously. The full rules for swapping are detailed in [section 8.8](#).
- 8.2.12. To TRASH a card is to move it from its current zone to its owner's discard pile. The full rules for trashing are detailed in [section 1.19](#).

8.3. Arranging and Rearranging Cards

- 8.3.1. To ARRANGE or REARRANGE an ordered set of 2 or more cards is to reposition them among their current locations. The words "arrange" and "rearrange" are synonymous.

- a. If a player is instructed to arrange 1 or fewer cards, instead that player does nothing.
- 8.3.2. When a player arranges a set of installed ice, that player carries out the arrangement by repeatedly choosing and swapping 2 pieces of ice. Each swap follows all the usual rules for swapping cards in [section 8.8](#).
- a. No checkpoints occur during the sequence of swaps. An ability that meets its condition while ice is being arranged will become pending at the end of resolving the instruction in which arranging ice is taking place.
- 8.3.3. When a player arranges cards from the top of their deck or their opponent's deck, that player sets those cards aside facedown, secretly puts them in the order of their choice, and returns them to the top of that deck. They do not declare which cards moved to which positions. All of the arranged cards become new objects after they are arranged, even cards that remain in the same location in the deck.

8.4. Drawing Cards

- 8.4.1. DRAWING cards is a process that moves cards from a player's deck to their hand.
- 8.4.2. To draw 1 or more cards, a player sets aside that many cards from the top of their deck facedown. The cards are then considered drawn, and abilities with trigger conditions related to cards being drawn can act on them. When resolving any such abilities is complete, the set aside cards are added to HQ.
- a. Abilities with a trigger condition that refers to cards being drawn can see the drawn cards in the set-aside zone. This is an exception to [rule 4.8.3](#).
 - b. Abilities that do not refer specifically to cards a player has drawn are subject to [rule 4.8.3](#), and only see the drawn cards move directly from the player's deck to their hand.
- 8.4.3. Abilities that resolve while drawn cards are set aside can move those cards or add new cards to that set.
- a. If a drawn card leaves the set-aside zone, it is no longer considered "drawn" and remains in its new location when the drawn cards are added to the hand.
 - b. If a drawn card is swapped with another card, the card swapped into the set-aside zone is now considered "drawn", can be manipulated by other abilities, and will be added to the hand with the other drawn cards.

Example: The Corp takes their mandatory draw while they have Daily Business Show and Raman Rai rezzed. They set aside Wraparound and Enigma from the top of R&D. After looking at those cards and the cards in Archives, the Corp uses Raman Rai to reveal the Enigma and a Tollbooth from Archives and swap them.

Then, the corp uses Daily Business Show to add Tollbooth to the bottom of the deck. Finally, the Corp adds Wraparound to HQ.

8.4.4. If an ability directs multiple players to draw cards at the same time, those players follow the procedure for drawing cards together. Abilities relating to all of those draws will become pending in the same reaction window.

8.4.5. Steps of Drawing N Cards

- a. Set aside N cards from the top of the drawing player's deck. The cards are now considered drawn and can be looked at by their controller.
- b. A checkpoint occurs.
- c. Add the set-aside cards to the player's hand.

8.5. Installing and Uninstalling Cards

8.5.1. INSTALLING an agenda, asset, ice, upgrade, hardware, program, or resource card is the act of placing that card into the play area. Events, operations, and identities are never installed.

- a. Cards and counters can also be installed onto other cards if either the card being installed onto or the card or counter being installed has an ability allowing this. The relationship created this way is called HOSTING and is discussed in [section 1.13](#). Some abilities also host cards without installing them.
- b. A card or counter is UNINSTALLED when it stops being installed for any reason.

8.5.2. Corp cards are installed facedown, unrezzed, and remain inactive until rezzed. Corp cards are installed into specific locations. See [section 4.6.6](#) for more information on servers and the locations of installed Corp cards.

- a. As part of installing a card, the Corp chooses which server will be the destination for that card. The Corp can either choose an already existing server as the destination or announces that the card will create a new remote server.
- b. The Corp always installs an agenda or asset into **the root of** a remote server.
- c. The Corp always installs an upgrade **into the root of a central or remote server**.
- d. Unless otherwise indicated, the Corp always installs ice into the outermost position protecting the destination server.

8.5.3. Cards in servers cannot be rearranged unless instructed by a card ability. To organize this hidden information for both players, it is important that the Corp observes the following rules for card orientation:

- a. Agendas, assets, and upgrades are always installed in a vertical orientation. An upgrade in **the root of** a remote server is installed in the same position as an

agenda or asset. The Runner should not be able to tell what type of card is installed in **the root of** a remote server by its position.

b. Ice is always installed in a horizontal orientation.

8.5.4. Runner cards are **normally** installed faceup, active. Runner cards are generally installed without designated locations.

8.5.5. Installing cards is an exception to [rule 9.12.2a](#). If an effect would install more than one card, then those cards are chosen and installed one at a time, following all of the steps of the installation process ([section 8.5.13](#)) for each card before choosing and installing the next. Each such installation is performed as a separate instruction. See [rule 9.11.4b](#).

Example: The Runner uses Mass Install to install three programs. The Runner can install Dhegdheer first, and then host one of the other two programs on Dhegdheer in order to reduce the install cost.

8.5.6. As part of the installation process, a player installing an agenda, asset, upgrade, ice, or program has the opportunity to trash like cards.

- a. **When installing an agenda or asset in the root of a remote server, or an upgrade in the root of any server, the Corp may first trash any other cards already installed in the root of that server. If the card to be installed is an asset or agenda, the Corp must trash any other asset or agenda from that server. If the card to be installed is a region, the Corp must trash any other region from that server.**
- b. When installing ice protecting a server, the Corp may first trash any other ice already installed protecting that server. Ice trashed in this way will not be counted when determining the install cost of the new ice.
- c. When installing a program, the Runner may first trash any number of programs already installed. They must do so if installing the new program would exceed their memory limit.

8.5.7. If the Corp trashes any cards during the installation process, the trashed cards are placed in Archives with the same faceup or facedown status they had while installed.

8.5.8. If the last card protecting a remote server **or in the root of that server** is uninstalled outside of the **process of installing another card**, then the server ceases to exist during the next checkpoint. See [section 10.3](#).

8.5.9. The Corp cannot destroy a server by installing cards. While the Corp is installing a card **with a remote server as its destination**, that server does not cease to exist during any checkpoints that occur before the install effect is complete. Even if the last card installed in **the root of** or protecting a remote server is trashed during the installation process, it is still considered the same server after the new card has been installed.

8.5.10. If a piece of ice is uninstalled while being approached or encountered, any open paid ability window is completed. Then the approach or encounter ends without completing

any further steps of that timing structure. The Runner passes the position the uninstalled ice had occupied, and the run continues. **See [section 6.2](#) for rules about positions during a run.**

- 8.5.11. Some types of cards have an install cost, which must be paid before the card can become installed, unless an ability indicates that this cost should be ignored.
- The install cost of a piece of ice is 1 credit for each piece of ice already installed protecting the destination server at the time the cost is to be paid. The install cost of a piece of hardware, program, or resource is indicated on the card. Assets, agendas, upgrades, and facedown Runner cards have no install cost. See also [section 1.16.5](#).
 - If a card has an install cost of X, the value for X is chosen by the player at the time the cost would normally be paid, according to any stipulations indicated in the card's text box. This value of X is maintained until the install effect is complete.
 - After the install cost is paid, a checkpoint occurs, as required by [rule 10.3.4](#).
- 8.5.12. Some upgrades have **an ability that specifies 1 or more central servers or "remote server" followed by the word "only"**. This is a restriction on the locations an upgrade can occupy that applies at all times, even if the upgrade is inactive. The Corp cannot choose the root of a **non-specified** server as the installation destination for such an upgrade, and such an upgrade cannot be swapped into the root of a **non-specified** server.

8.5.13. Steps of Installing a Card

- Place the card into the play area with the same faceup or facedown status it will have when the installation is complete. It is not yet installed or active.
- Choose and declare the install destination appropriate to the card that will be installed, including any host relationships, if applicable.
- Trash like cards as described in [section 8.5.6](#).
- Pay the appropriate install cost. (The cost-paid checkpoint then occurs.)
- If the card is to be the first card in **the root of** or protecting a new remote server, that server is created. Move the card to the chosen install location. It becomes installed. If the card is faceup, it becomes active.
- Any "When installed..." abilities that apply to the installation meet their trigger conditions, including those on the installed card, and the install effect is complete.

8.6. Playing Events and Operations

- 8.6.1.** PLAYING an event or operation is the act of placing that card into the play area. Agendas, assets, ice, upgrades, hardware, programs, and resources, and identities are never played.
- 8.6.2.** Normally, playing an operation or event requires paying its play cost. Abilities that initiate playing a card may ignore the play cost or require it to be paid.
- If a card has a play cost of X, the value for X is chosen by the player at the time the play cost is paid, according to any stipulations written on the card. This value of X is maintained while resolving the event or operation.
 - After the play cost is paid, a checkpoint occurs, as required by [rule 10.3.4](#).
- 8.6.3.** Playing cards is an exception to [rule 9.12.2a](#). If an effect would play more than one card, then those cards are chosen and played one at a time, following all of the steps of the play process ([section 8.6.6](#)) for each card before choosing and playing the next. Each such card is played as a separate instruction. See [rule 9.11.4b](#).

Example: The Corp uses Subcontract to play Hedge Fund and another operation. The Corp can use the credits gained from Hedge Fund to pay for the second operation.

- 8.6.4.** Operations and events can create lingering effects. Once those effects exist, they become independent of the source operation or event and the card is considered fully resolved and ready to be trashed. See [section 9.10](#).

Example: Test Run creates a delayed conditional ability that will resolve at the end of the turn, but Test Run itself is considered fully resolved and is trashed once it finishes installing a program.

- 8.6.5.** An operation or event that initiates a timing structure remains active in the play area for the duration of that timing structure. It is not considered fully resolved until the timing structure is completed and any abilities on the card that meet their trigger conditions when the timing structure ends are resolved.

8.6.6. Steps of Playing an Event or Operation

- Place the card faceup into the play area. It is not installed and not yet active.
- Pay the play cost. (The cost-paid checkpoint then occurs.)
- The card becomes active.
- Any “When played...” abilities that apply meet their trigger conditions.
- A checkpoint occurs.
- Resolve the play abilities of the card.

- g. Trash the card **if it is still in the play area**.
- h. Any “After you resolve...” abilities that apply meet their trigger conditions.
- i. The play effect is complete.

8.7. Searching for Cards

- 8.7.1. If a player is instructed to SEARCH a zone for one or more cards, they look at all the cards in that zone for cards matching the criteria of the search.
 - a. If a player is instructed to search a hidden or secret zone, they are temporarily allowed to look at those cards until the search is complete, so long as they maintain that zone as hidden or secret as necessary to the other player.
- 8.7.2. While a player is searching, they may FIND an appropriate number of cards matching any criteria of the search by taking them from the zone being searched and setting them aside facedown.
 - a. A card cannot be found for a search unless it both matches the specified criteria of the search and can be acted upon by any other effects referencing the found card in the ability that initiated the search.

Example: The Runner uses Artist Colony to search for a card and install it. They cannot find an event with the search because events can never be installed. They also cannot find a program, hardware, or resource that they cannot afford to install.

- b. The searching player is not required to reveal any found cards unless instructed to do so by the ability that initiated the search. Found cards are not revealed until resolution of the ability that initiated the search resumes, pending any necessary shuffling.
 - c. If a player is searching for a set number of cards without any specified criteria, they must find that many cards from that zone or all the cards in that zone if there are not enough.
 - d. If a player is searching a deck for one or more cards with specified criteria, they may choose to fail to find anything, even if one or more cards matching the criteria exist in that deck.
 - e. If a player is directed to search a zone for cards “with different names,” each card that player finds must have a different English name from every other card found for that search.
- 8.7.3. Once a search through a deck is complete, whether or not any cards are found, the deck must be immediately reshuffled before continuing to resolve any remaining effects from the ability that initiated the search. The shuffling takes precedence over anything that would be done with the found card(s), as well as any chain reactions that occur as a result of the search, finding, or remaining abilities.

Example: The Corp is playing Near-Earth Hub and uses the ability on Tech Startup to search their deck for an asset and install it. After finding the asset, R&D must be immediately shuffled before the Corp installs the asset or triggers Near-Earth Hub's ability.

- 8.7.4. Once the search is complete, and any shuffling necessary has occurred, resolution of the ability that initiated the search resumes. If no cards were found during the search, any effects referencing found cards fail to resolve, but as much of the remaining effects resolve as possible.
- 8.7.5. If the trigger condition of a conditional ability involves a search, that ability becomes pending after the search is completed and any shuffling is performed, but before the found cards are acted on.

8.8. Swapping Cards

- 8.8.1. To SWAP two cards is to exchange their locations.
- 8.8.2. A card can only ever be swapped into a location it is normally allowed to occupy. When a player resolves a swap effect, they must observe any applicable game rules or card abilities that would affect that card in its final destination. If a swap effect would resolve while there are no legal exchanges possible, then that effect does nothing.

Example: When the subroutine on Metamorph resolves, the Corp cannot choose to swap an agenda and an upgrade such that the agenda would end up in the same remote server as an asset.

- 8.8.3. If two installed cards are swapped, each card moves to the other's location simultaneously. The two cards remain installed throughout the process of swapping, and do not otherwise affect any other part of the game state.
- 8.8.4. If two cards are swapped between zones, each card moves to the zone the other is in simultaneously.
 - a. Each of the swapped cards enters its destination zone in the same state that a card would normally enter that zone. A card swapped into Archives enters Archives facedown unless it was visible to the Runner at the time of the swap, a Corp card swapped into the play area enters unrezzed, etc.
 - b. If only one of the cards to be swapped is installed, then it becomes uninstalled as it moves to the destination zone. Any cards or counters hosted on it are trashed. The other card becomes installed in the exact position the first card occupied, maintaining any specific location (such as server, ice position, or host).
 - c. If two agendas are swapped between the players' score areas, any hosted cards or counters remain hosted on their respective host agendas.

9. Abilities

9.1. General

- 9.1.1. An ABILITY is an independent unit of text on a card or counter, a basic action, **or the basic trash ability**.
- a. All rules text on a card or counter is part of an ability.
 - b. The basic actions are defined in [section 5.2.7](#) and [section 5.2.8](#).
 - c. **The basic trash ability is defined in [section 7.1.5](#).**
 - d. An ability's text can include other abilities that it could grant to cards or counters or introduce directly to the game state. See also [section 9.10](#).
 - e. Each ability is categorized as either a static ability ([section 9.4](#)), a paid ability ([section 9.5](#)), a conditional ability ([section 9.6](#)), a play ability ([section 9.7](#)), **or a subroutine ([section 9.8](#))**.
 - f. Abilities marked with [interrupt] or that include the words "prevent" or "avoid" in their instructions have special timing rules that differ from other abilities. See [section 9.9](#).
- 9.1.2. To RESOLVE AN ABILITY means to resolve each of that ability's instructions in the order they appear. If an ability contains more than one instruction, a checkpoint occurs between each consecutive instruction. To RESOLVE AN INSTRUCTION means to carry out that instruction.
- a. While resolving an ability, other abilities can meet their conditions. When this happens, a "chain reaction" is created. The current instruction finishes resolving, then more recent abilities fully resolve before the next instruction of the original ability. If any other abilities meet their conditions while resolving the "chained" abilities, then another "chain" is created before continuing to resolve the previous chained abilities. Resolve each set of chained abilities one at a time, from the most recently met condition to the oldest, before continuing with the original ability that started the chain reaction.
- Example: The Runner's identity is Armand "Geist" Walker. They access a Snare! from R&D with only 2 cards in their grip. Before taking a tag or suffering any net damage, the Runner triggers Decoy's ability in order to avoid the tag. Using the Decoy meets the trigger condition of Geist's ability. As this is the most recent ability to meet its trigger condition, Geist's ability must resolve first, before Decoy avoids the tag and before Snare! finishes resolving. The Runner draws a card from Geist, avoids the tag from Snare! with Decoy, and then finally suffers (and survives) the 3 net damage from Snare!.*
- 9.1.3. The SOURCE of an ability is the card, counter, or game rule that originated it.

- a. Each card is the source of its own printed abilities.
- b. **If something grants an ability to an object, the source of the granted ability is that object, not whatever granted the ability.**

Example: Dedicated Processor is hosted on Corroder, granting it the ability “2[c]: +4 strength.” The source of this ability is Corroder, not Dedicated Processor.

- c. The source of **an ability that is being maintained by a lingering effect** is the **object** that created **that lingering effect**. See [section 9.10](#).

- 9.1.4. Each **paid ability, conditional ability, and subroutine** becomes independent of its source at a certain point before it resolves, as described in [rule 9.5.4](#), [rule 9.6.12](#), and [rule 9.8.9](#), **respectively**. If the source changes zones after that point, the ability cannot act on the source.

Example: The runner plays Compile, uses it to install Mayfly during the resulting run, and breaks subroutines with Mayfly during that run. When the run ends, delayed conditional abilities from both Mayfly and Compile become pending. The Runner decides to resolve Compile first, adding Mayfly to the bottom of their stack. The Runner then resolves the ability from Mayfly, which instructs them to trash its source program. Because Mayfly has changed zones, the copy of it on the bottom of the stack is a new object. With nothing to trash, the ability from Mayfly does nothing.

- 9.1.5. Whereas an ability, instruction, or declaration is made up of text, an EFFECT is what happens in the game because of that text.
- a. Any of a non-static ability’s effects that apply beyond the duration of that ability’s resolution become independent of that ability and its source. The game engine takes responsibility for managing these LINGERING EFFECTS directly, and deletes them from the game state as their durations expire. See [section 9.10](#).
 - b. When determining whether a certain ability has the potential to change the game state, look only at the expected effects of the ability. Do not consider its costs or restrictions, and do not consider other abilities that could become pending or relevant because of triggering or resolving the ability. See [rule 1.2.5](#) of the Golden Rules.
- 9.1.6. Any time a player chooses to resolve an optional ability or an optional part of an ability, the player is considered to be USING that ability and the source card of that ability. Players do not “use” abilities that are entirely mandatory.
- a. A paid ability is considered used once the trigger cost has been paid. See [section 9.5.6](#) for the steps of using a paid ability.
 - b. A conditional ability is considered used once the relevant optional effects of the ability have been resolved by the controller of the ability. See [section 9.6.14](#) for the steps of triggering and resolving conditional abilities.

- c. If an ability allows a player to spend or move counters, that ability's source card is considered used **at the time the player pays a cost by spending or moving those counters, even if the counters are spent or moved from cards other than the card that is the source of that ability.**

Example: The Runner spends the credit hosted on Cyberfeeder to pay the trigger cost of Mimic's ability. Because the Runner triggered Mimic's paid ability and Cyberfeeder's ability allowed the credit to be spent, both Mimic and Cyberfeeder have been used.

- d. A player can only use an ability if its effect has the potential to change the game state. See [rule 1.2.5](#).

9.1.7. Abilities are ACTIVE if they are eligible to be resolved or apply to the current game state, and INACTIVE otherwise. By default, a card's abilities are active if and only if that card is active (see [section 1.8.3](#)). Abilities of counters are active for as long as the counter exists.

9.1.8. Some types of abilities are active even while their source card is inactive.

- a. Conditional abilities that meet their conditions when their source card is accessed are active even while that card is inactive.
- b. Abilities stating that they are active in a particular zone are active in that zone. Abilities that can only ever meet their conditions in a particular zone are active in that zone. Abilities that can only affect the game state from a particular zone are active in that zone. When determining whether these stipulations apply, refer only to the game rules, not to any other effects that may be changing how cards move between zones.

Example: I've Had Worse has an ability that meets its trigger condition when it is trashed due to damage. Normally, this can only occur by moving I've Had Worse from the grip to the heap. Therefore, this ability is active in the heap. However, if I've Had Worse is trashed, but the Corp uses Skorprios Defense Systems to remove it from the game instead of adding it to the heap, the ability is not active.

- c. Abilities that modify when or if their source card can be played, installed, or rezzed are active even while that card is inactive.
- d. Abilities that modify the cost to install, rez, or play their source card are active even while that card is inactive.
- e. Abilities that modify the advancement requirement of their source card or create an additional cost to steal their source card are active even while that card is inactive.
- f. Abilities allowing a card to be advanced are active even while that card is unrezzed.
- g. If an active card is trashed, an ability of that card with a trigger condition that is met by this trashing remains active in its owner's discard pile until any**

corresponding instances of the ability resolve. See [rule 9.6.2](#) for information about instances of conditional abilities.

Example: The Runner uses Singularity to simultaneously trash a rezzed Hostile Infrastructure and 2 upgrades. Those cards are moved to Archives and become inactive, but Hostile Infrastructure's ability remains active, and 3 instances will become pending, each doing 1 net damage when it resolves. Once all those instances are resolved, Hostile Infrastructure's ability stops being active.

- h. If a piece of ice is encountered while it is not installed, its subroutines are active during that encounter.

Example: The Runner accesses Archangel from HQ, and the Corp uses its ability to force the Runner to encounter it. Archangel's subroutine is active and can resolve during that encounter.

- i. **Persistent abilities can sometimes remain active after their source card is trashed. See [section 9.12.5](#).**

9.2. Timing and Priority

- 9.2.1. The ACTIVE PLAYER is the player whose turn it is. The other player is the INACTIVE PLAYER.
- 9.2.2. A TIMING STRUCTURE is a unit of the game in which a prescribed sequence of steps progress the game forward.
 - a. The Corp's turn and the Runner's turn are timing structures, as are each of the 3 phases of the turns. See [section 5](#).
 - b. A run is a timing structure, as are each of the 6 phases of a run. See [section 6](#).
 - c. **Breaching a server** is a timing structure. See [section 7.3](#) and [section 7.5](#).
 - d. **Accessing a card** is a timing structure. See [section 7.1](#) and [section 7.2](#).
 - e. **Other procedures with prescribed steps, such as installing a card or resolving a trace attempt, are not timing structures.**
- 9.2.3. PRIORITY is a player's opportunity to act and make certain game choices. No more than one player can have priority at any given time.
- 9.2.4. A PRIORITY WINDOW is a general term for a timing step in which one or both players receive priority. Priority windows OPEN for different purposes throughout the game. When a priority window CLOSES, the game continues to the next timing step.
 - a. All priority windows give the player with priority the option to choose a relevant ability they control (if there are any) as the next ability for the game to resolve. This is often referred to as TRIGGERING the chosen ability. Some priority windows also give players other options.

- b. Except during action windows, the player with priority has the option to PASS in addition to the available options defined by that priority window. When a player passes, the game progresses to the next step, either by giving priority to the other player or by closing the priority window. Each type of priority window defines the method and resolution of passing.
 - c. Unless otherwise noted, the player with priority receives priority again after resolving one of their available options. That player will continue to receive priority until they pass.
 - d. While a priority window is open, another nested priority window can open as well. This nesting allows the game to resolve “chain reactions,” as discussed in [rule 9.1.2a](#). The most recently opened priority window is always resolved before returning to an earlier priority window.
 - e. Whenever a player receives priority during a priority window, a checkpoint occurs immediately before that player may act. See [section 10.3](#).
- 9.2.5. The types of priority windows are action windows, paid ability windows, reaction windows, interrupt windows, **and mid-access ability windows**.
- 9.2.6. An ACTION WINDOW is a priority window that opens during a player’s action phase if they have unspent clicks.
- a. An action window gives only the active player priority.
 - b. During an action window, the active player must take an action. [Section 5.2](#) discusses actions. This type of window does not give the option to pass.
 - c. After the active player takes an action, the action window closes and the game moves to the next timing step. The player does not receive priority again.
 - d. There are two timing steps during which an action window may open. The Corp has an action window in [step 5.6.2b](#) of their action phase. The Runner has an action window in [step 5.7.1f](#) of their action phase.
- 9.2.7. A PAID ABILITY WINDOW is a priority window that opens throughout the game’s timing structures to allow players to trigger paid abilities, rez cards, or score agendas.
- a. Paid ability windows give both players priority, starting with the active player. When a player passes, the other player receives priority. Players continue to exchange priority until a player who receives priority from their opponent passes without resolving any other option available to them. Once this happens, the paid ability window closes.
 - b. During every paid ability window, the player with priority has the option to trigger an active paid ability they control. **Players cannot trigger actions, interrupts, or mid-access abilities in a paid ability window**. See [section 9.5](#). Within this

document, paid ability windows are marked with the symbol (P) to denote the option to trigger a regular paid ability.

- c. During some paid ability windows, the Corp has the option to rez an asset or upgrade while they have priority. Within this document, the symbol (R) denotes a paid ability window in which cards can be rezzed. See [section 8.1](#).
- d. During some paid ability windows, the Corp has the option to score an agenda while they have priority. Within this document, the symbol (S) denotes a paid ability window in which agendas can be scored. See [section 1.17](#).
- e. During the paid ability window at [step 6.9.2b](#) of the Approach Ice Phase of a run, the Corp has the option to rez the piece of ice the Runner is approaching while they have priority.
- f. The player with priority during a paid ability window may use any of the options available to them any number of times in any combination and order until they decide to pass, so long as they are allowed and the player pays any and all costs to use each option. Each option must be fully resolved before another is chosen. A player is not obligated to resolve any of the options available to them, except they must pass.
- g. Paid ability windows occur throughout the timing steps of turns and runs. [Section 5.6](#), [section 5.7](#), and [section 6.9](#) detail those steps and indicate which options are available in which windows. [Section 11](#) also contains a summarized version of these steps.

9.2.8. A REACTION WINDOW is a priority window that opens whenever one or more active conditional abilities become pending by meeting their conditions. [Section 9.6](#) discusses conditional abilities.

- a. Each reaction window is associated with the fixed set of conditional abilities that met their conditions just before the window opened. If other abilities become pending during a reaction window, a separate reaction window opens to handle the new abilities. See [rule 9.1.2a](#).
- b. Reaction windows give both players priority, starting with the active player. When the active player passes, the inactive player receives priority. When the inactive player passes, the reaction window closes.
- c. During a reaction window, the player with priority has the option to trigger a pending conditional ability they control that is associated with that window.
- d. The player with priority during a reaction window may trigger their pending abilities in any order until they decide to pass; they do not need to trigger mandatory abilities before optional ones. Each ability they trigger must be fully resolved before another is chosen.
- e. The player with priority cannot pass if they control any pending mandatory abilities. They may pass with optional conditional abilities still pending, in which case those

abilities lose their pending status without being triggered. [Section 9.6.9](#) describes the differences between mandatory and optional conditional abilities.

- f. If a reaction window opens due to a timing structure beginning, and during that reaction window the timing structure ends (e.g. by an effect moving the game to another timing point past the end of the structure), then the reaction window immediately closes. All remaining abilities associated with the window lose their pending status without being triggered, even if they are mandatory abilities.

Example: The Runner has a Femme Fatale installed and chose a Tollbooth with its “when installed” ability. When the Runner encounters the Tollbooth, they pay 1[c] to bypass the Tollbooth with Femme Fatale. Because the resolution of Femme Fatale’s ability causes the encounter to end, the pending ability from Tollbooth cannot be triggered. The Runner does not pay 3[c], and the run does not end.

9.2.9. An INTERRUPT WINDOW is a priority window that opens just before an instruction would resolve when one or more players have abilities that could modify that imminent instruction. See [section 9.9](#) for rules about interrupt abilities.

- a. Each interrupt window is associated with the single imminent instruction being modified by the abilities triggered during the window, and with a fixed set of conditional ability interrupts determined as the window opens.
- b. As an interrupt window opens, before players receive priority, the expected effects of the imminent instruction are determined, any applicable replacement effects are applied, and then relevant conditional abilities become pending. See [section 9.9.4](#).
- c. Interrupt windows give both players priority, starting with the active player. When a player passes, the other player receives priority. Players continue to exchange priority until a player who receives priority from their opponent passes without resolving any other option available to them. Once this happens, the interrupt window closes.
- d. During an interrupt window, the player with priority has the option to trigger an interrupt ability they control that is relevant to the imminent effect. See [section 9.9.3](#).
- e. The player with priority during an interrupt window may trigger their abilities in any order until they decide to pass. They do not need to trigger mandatory abilities before optional abilities or trigger conditional abilities before paid abilities. Each ability they trigger must be fully resolved before another is chosen.
- f. The player with priority cannot pass if they control any pending mandatory abilities that are still relevant to the imminent effect. They may pass with optional conditional abilities still pending. ([Section 9.6.9](#) describes the differences between mandatory and optional conditional abilities.) Once a player has passed, if they receive priority again, they may continue to trigger paid abilities and pending conditional abilities that are relevant. As the interrupt window closes, any remaining pending conditional abilities lose their pending status.

- 9.2.10. A MID-ACCESS ABILITY WINDOW is a priority window that opens while the Runner is accessing a card.**
- a. A mid-access ability window gives only the Runner priority.**
 - b. During a mid-access ability window, the Runner may use a mid-access ability or may pass. In addition to any mid-access abilities available from cards, the Runner may use the basic trash ability (see [section 7.1.5](#)).**
 - c. After the Runner uses a mid-access ability or passes, the mid-access ability window closes and the game moves to the next timing step. The Runner does not receive priority again.**
 - d. The only point at which a mid-access ability window occurs is at [step 7.2.2](#) of accessing a card.**

9.3. Interpreting Card Text

- 9.3.1. Text is classified into conditions, restrictions, instructions, declarations, **and ability flags**. The rules text of an ability can contain text in any combination of these classes, except that no ability contains both instructions and declarations.
- 9.3.2. A **CONDITION** is a unit of text that stipulates a requirement for when an ability's effects are allowed to apply.
- a. A **COST CONDITION** describes a cost (either a nested cost or a trigger cost) that a player must pay to apply an effect. Costs are discussed in detail in [section 1.16](#).
 - b. A **TRIGGER CONDITION** describes a change in game state that must occur for an effect to apply. Trigger conditions often begin with words such as "if", "when", "whenever", or "after", or with ordinal phrases such as "the first time".
 - c. A **STATIC CONDITION** describes a property of the game state that must be true for an effect to apply. Static conditions often begin with words such as "if", "during", or "while".
- 9.3.3. A **RESTRICTION** is a unit of text that applies one of a specific set of constraints for a card to be played or an ability to be used. If a restriction appears in an ability, it applies to the entire ability regardless of whether it is written before, after, or in the middle of the ability's other text. Many restrictions fall under [section 9.1.8](#), meaning they are active even while their source is inactive.
- a. Text modifying the cost to play, install, rez, or steal the card it appears on is a restriction, including text that adds an additional cost.
 - b. Constraints on when or where a card can be installed, rezzed, played, or scored are restrictions.
 - c. Limits on when, where, or how often an ability can be used are restrictions.

- d. Constraints on how a cost can be paid are restrictions.
- e. If a card's text limits what cards it can host, the text describing that limit is a restriction. See also [rule 1.13.5b](#).
- f. Definitions for or constraints on a variable are restrictions.

*Example: Some abilities dictate a value for X, such as "X is the number of rezzed NEXT ice." Some abilities with X in their cost constrain the value that can be chosen for X, such as "X **must be** equal to or less than the number of tags the Runner has." These statements are restrictions.*

- g. Stipulations that an effect or part of an effect cannot be prevented are restrictions.
- h. Any text not falling under one of the above categories is not a restriction.

9.3.4. An INSTRUCTION is a statement or command that is resolved at a specific time and applies immediate effects to the game state. (This can include creating new lingering abilities or effects that will continue to apply over time; see [section 9.10](#).)

- a. Instructions can originate from a game rule or from the text of an ability. An instruction in an ability resolves when that ability resolves (see [rule 9.1.2](#)). An instruction in the game rules resolves during the timing step(s) when it appears. **[Section 9.11](#) explains how to determine the boundaries between consecutive instructions.**
- b. If an instruction requires any targets, players announce those targets before that instruction becomes imminent. See [section 1.15](#).
- c. Each instruction is carried out as an atomic unit and cannot be interrupted once it begins to resolve. The procedure for carrying out an instruction can be altered by other effects such as interrupts, but only before the instruction begins to resolve. See [section 9.9](#).
- d. Other than choosing targets, carry out the steps of a single instruction in the order they are written.
- e. Instructions in an ability can create effects that last beyond the resolution of the ability. See [section 9.10](#).

9.3.5. A DECLARATION is a statement describing an effect on components or game rules that is applied continuously. A declaration applies its effects as long as it is active.

9.3.6. An ABILITY FLAG is a keyword or symbol that appears at the beginning of an ability. Ability flags are separated from the main text of the ability by an arrow (→). Each ability flag changes the rules for how the flagged ability functions.

- a. There are four ability flags: access, interface, [interrupt], and persistent.**

- b. The “access” flag appears on Runner card paid abilities and affects the timing for triggering those abilities. The Runner can use abilities with this flag only during the mid-access ability window at [step 7.2.2](#) of accessing a card. See [section 9.2.10](#).
- c. The “interface” flag appears on icebreaker paid abilities and affects the timing for triggering those abilities. The Runner can use abilities with this flag only during an encounter, and only if the ability’s source is an icebreaker with strength that equals or exceeds the strength of the encountered ice. See [section 3.9.5](#).
- d. The interrupt flag ([interrupt]) appears on paid and conditional abilities and affects the timing for triggering those abilities. Players can use abilities with this flag only during interrupt windows, and only if the ability is relevant to the imminent instruction. See [section 9.9](#).
- e. The “persistent” flag appears on Corp card abilities and changes the rules about when those abilities are active. An ability with this flag can persist until the end of the run after its source card is trashed. See [section 9.12.5](#).

9.3.7. The **five** types of abilities can be identified by the types of text they are made of.

- a. Static abilities ([section 9.4](#)) can contain declarations, restrictions, and conditions, but never instructions. Static abilities are the only ability type that can contain declarations.
- b. Paid abilities ([section 9.5](#)) can always be identified by their formatting: a cost condition in bold text, a colon (:), then the remainder of the ability. The text after the colon can contain conditions, restrictions, and/or instructions.
- c. Conditional abilities ([section 9.6](#)) always contain a trigger condition or static condition and at least one instruction. They can also contain restrictions, other instructions, or nested conditions. The primary condition is often written with a timing-related word or phrase like "when", "after", "the first time", and so on, and is usually the first part of the ability’s text. Some older cards write the condition at the end of the ability or use "if" to indicate the condition.
- d. Play abilities ([section 9.7](#)) are the abilities on events and operations that are not paid, conditional, or static abilities. They can contain conditions, restrictions, and/or instructions.
- e. **Subroutines ([section 9.8](#)) always begin with the [sub] symbol, and can contain conditions, restrictions, and/or instructions.**

9.4. Static Abilities

- 9.4.1. A STATIC ABILITY is an ability that continuously affects the game as long as it is active. Static abilities are the only type of ability that can contain declarations, and they do not resolve or have associated priority windows.

- 9.4.2. If a static ability contains a static condition, then the constrained parts of the ability apply to the game state only if that condition is met.
- 9.4.3. Static abilities can include restrictions that apply to their source card. These restrictions are often active even while the card is inactive. See [section 9.1.8](#).
- 9.4.4. The effects of static abilities do not have durations and cannot directly create lingering effects (see [section 9.10](#)).

Example: The runner controls Puffer with a hosted Gebrselassie. Gebrselassie changes the durations of abilities affecting its host's strength, so if the Runner uses Puffer's paid ability to give it +1 strength, that increase will last for the remainder of the turn. Gebrselassie does not affect Puffer's static ability that increases its strength for each hosted power counter, so if the Runner spends [click] to remove a power counter, then the increase in strength from Puffer's static ability will immediately be lost.

- 9.4.5. Static abilities that modify a value maintain the same restrictions and specifications that were present on the original value.

Example: The Cleaners has a static ability that gives every instance of meat damage done by the Corp +1 to the amount of the damage. Part of Flare's effect does 2 meat damage that can't be prevented. If this effect resolves while The Cleaners is active, then all 3 points of meat damage are unpreventable.

9.5. Paid Abilities

- 9.5.1. A PAID ABILITY is an ability players trigger at-will, during appropriate priority windows. In order to use a paid ability, the controller of that ability must pay its TRIGGER COST. Paid abilities are always written with the trigger cost first, followed by a colon (:), followed by the remainder of the ability's text.
- 9.5.2. A player can trigger paid abilities they control while they have priority in an appropriate priority window.
 - a. If the trigger cost of a paid ability **begins with a [click] symbol**, the ability is an ACTION. A player can use one action paid ability during each action window on their turn. See [section 9.2.6](#) and [section 5.2](#).
 - b. If the [interrupt] **flag** appears before a paid ability's trigger cost, that ability is an interrupt. A player can use interrupt paid abilities during appropriate interrupt windows. Each such ability can be used as many times as its effects could apply to the imminent instruction. See [section 9.9](#).
 - c. **If the "access" flag appears before a paid ability's trigger cost, that ability is a mid-access ability. The Runner can use up to 1 mid-access ability during the mid-access ability window at [step 7.2.2](#) of accessing a card.**

- d. If the “interface” flag appears before a paid ability’s trigger cost, that ability is an Interface ability and is subject to the restrictions of icebreakers discussed in [section 3.9.5](#). Interface abilities do not have a special priority window type.
- e. A paid ability that is not an action, interrupt, or mid-access ability can be used during a paid ability window. Each such ability can be used an unlimited number of times as long as its cost is paid each time and any restrictions specified by its effect are observed.

9.5.3. Paid abilities are always optional. A paid ability and its source are considered used when the ability’s trigger cost is paid.

- a. Mid-access abilities the Runner is forced to trigger still count as optional abilities. See also [rule 7.1.5c](#) and [rule 7.1.5d](#).

Example: The Corp resolves the subroutine on Wendigo, prohibiting the Runner from using their installed Imp for the remainder of the run. Later, the Runner accesses an installed Mumbai Virtual Tour. Even though Mumbai Virtual Tour forces the Runner to trash it if able, Imp’s ability is still optional for purposes of Wendigo’s prohibition. Since Imp’s ability cannot be used at all this run, Mumbai Virtual Tour cannot compel the Runner to spend one of its hosted virus counters. If the Runner can pay the cost of another mid-access ability, such as the basic trash ability, the Runner must use that ability. If not, the Runner will not be able to trash Mumbai Virtual Tour at all.

9.5.4. Once a player pays the trigger cost of a paid ability, that ability becomes independent of its source and [rule 9.1.4](#) applies to it.

9.5.5. **If the trigger cost of a paid ability uninstalls or forfeits the source of that paid ability, and the effects of that paid ability refer to or act on cards or counters hosted on that source, set aside any hosted cards and counters** as the trigger cost is paid. The set-aside cards or counters are still considered “hosted” for purposes of this ability, but they are not trashed due to not having a host for as long as the ability is resolving. When the ability finishes resolving, if any of those cards or counters are still set aside, they are trashed during [step 10.3.1f](#) or [step 10.3.1g](#) of the next checkpoint, as appropriate. Other abilities cannot interact with these cards or counters while they are set aside (see [rule 4.8.3](#)).

Example: The Runner has Fermenter installed with 4 hosted virus counters. They use its paid ability, which has a cost that includes trashing Fermenter. As the Runner pays this cost, they set aside the 4 counters. When the ability resolves and refers to the number of hosted virus counters, it includes the counters that were set aside, so the Runner gains 8 credits. When the ability is finished resolving, the counters are returned to the bank.

Example: The Runner uses the [trash] ability on Street Peddler. They pay the trigger cost and set the 3 hosted cards aside. When the ability resolves, the Runner installs one of the set-aside cards. Since the other two cards are still set aside, the next checkpoint trashes them just as in any other case of cards that were hosted on a card that is no longer installed. No other abilities

are able to tell that the cards were set aside: they are treated as having been installed or trashed from their previous location in the play area.

*Example: This example describes a situation similar to the previous examples, but covers the sequence of steps the game carries out in a higher level of detail. The Corp has priority in a paid ability window, and chooses to trigger the [trash] ability on Reconstruction Contract. As they pay the trigger cost, moving Reconstruction Contract to Archives, they set aside the advancement **counters** hosted on Reconstruction Contract. Next, a checkpoint occurs following the cost having been paid. After that checkpoint and any corresponding reaction window, the [trash] ability's instruction is ready to become imminent. The Corp chooses a card they can advance to be the instruction's target, and an interrupt window opens. After the interrupt window (assuming the effects of the instruction are not prevented), the Corp moves the set-aside **counters** to the target card.*

9.5.6. Steps of Using a Paid Ability

- a. Announce the intent to trigger the paid ability.
- b. Pay the trigger cost. The ability and its source are considered used. "When used" abilities meet their trigger conditions. (The cost-paid checkpoint then occurs.)
- c. Announce any targets for the ability's first instruction. That instruction then becomes imminent.
- d. An interrupt window occurs, during which abilities can modify, prevent, or avoid the imminent effects.
- e. Resolve the instruction, applying any changes to its effects from interrupts that were resolved.
- f. A checkpoint occurs.
- g. If there are more instructions to resolve, announce any targets for the next instruction. That instruction then becomes imminent. Return to (d).
- h. Resolution of the paid ability is complete.

9.6. Conditional Abilities

- 9.6.1. A CONDITIONALABILITY is an ability that a player can or must trigger at a specific point in the game. Conditional abilities always include a primary condition and one or more instructions, but they have no special syntax requirements.
 - a. The primary condition of a conditional ability is usually a trigger condition, but can also be a static condition.
 - b. The primary condition of a conditional ability is often, but not always, written at the front of the ability. See [rule 9.3.7c](#) for details about conditional ability syntax.

- 9.6.2. When a conditional ability has met its condition, one or more INSTANCES of that conditional ability is created PENDING in the next reaction window that opens. Each instance of a conditional ability is a separate copy of that ability that resolves independently of the others. An instance of an ability with the pending status is waiting for its controller to trigger it.
- 9.6.3. A conditional ability with a static condition can only have one instance at a time.
- 9.6.4. A conditional ability with a trigger condition can have multiple instances.
- If a trigger condition is met again while an instance of the corresponding ability is already pending, imminent, or resolving, a new instance can still become pending in the next reaction window.
 - If the trigger condition of an ability is met more than once between consecutive checkpoints, multiple instances of that ability become pending in the next reaction window.

Example: The Runner plays Singularity, trashing 3 Corp cards simultaneously. Since Hostile Infrastructure's trigger condition is met separately for each card trashed, the next checkpoint handles all of those occurrences, and 3 instances of the ability become pending in the same reaction window.

Example: The Runner controls Blackguard and plays Satellite Uplink, exposing 2 cards in a single instruction. In the next checkpoint, two instances of Blackguard's ability become pending, one corresponding to each card that was exposed.

- 9.6.5. The trigger condition of most conditional abilities describes an occurrence that allows it to become pending. These rules about trigger conditions also apply to conditional abilities with static conditions, except as described in [section 9.6.7](#).
- During each checkpoint, the game checks whether any conditional abilities have met their trigger conditions. If any have, a reaction window opens to resolve those abilities (See [section 10.3](#)). This reaction window is fully resolved and closed before the game proceeds to the next instruction of the original effect or game rule.
 - A conditional ability can only recognize its trigger condition occurring if the ability is active at the time the trigger condition resolves. If a conditional ability becomes active after the point when its trigger condition was met, that ability does not become pending, even if other abilities sharing that trigger condition are still pending.
 - A conditional ability can only recognize its trigger condition occurring if all the stipulations and requirements listed in the trigger condition are met at the moment that the trigger condition would occur.

Example: The entire trigger condition on Quantum Predictive Model is "If the Runner is tagged when Quantum Predictive Model is accessed". Quantum Predictive Model can only meet its trigger condition if the Runner is tagged at the time the access occurs.

Even if Casting Call is hosted on Quantum Predictive Model, the tag will be given after the access begins, and no ordering of the abilities will allow Quantum Predictive Model to meet its trigger condition.

- d. Additional stipulations and requirements listed in the instructions of a conditional ability are checked as part of the ability's effects and are not part of the ability's trigger condition. For the effects under the scope of this kind of condition to resolve, the condition only needs to be met when the relevant instructions resolve. This can occur even if some or all of those conditions were not met at the time the ability met its trigger condition.

Example: If a Runner with 1 link has both Underworld Contact installed and The Supplier installed hosting a Dyson Mem Chip, both Underworld Contact and The Supplier meet their trigger conditions at the same time. The Runner can trigger The Supplier first, installing the Dyson Mem Chip, thus allowing Underworld Contact to recognize that the Runner has 2 link and give them 1 credit, even though the Runner did not have 2 link at the time the Underworld Contact became pending.

- e. The phrase "If successful" in reference to a run is a trigger condition with specific rules. See [section 6.7.4](#). The phrases "If successful" and "If unsuccessful" are also frequently used as trigger conditions relating to the results of a trace. See [section 10.8](#).

- 9.6.6. Trigger conditions look for an instantaneous change in the game state. The next checkpoint after that change takes place marks an instance of the ability as pending for each occurrence of the change that was looked for.
- 9.6.7. Conditional abilities with a static condition instead of a trigger condition describe an effect that must be performed repeatedly, if possible, while the condition is true. Such abilities usually attempt to end their own repetition by uninstalling their source.
 - a. During each checkpoint, the game checks whether any conditional abilities with static conditions should become pending, alongside conditional abilities with trigger conditions. However, there are further requirements for this kind of ability to be marked pending.
 - b. A conditional ability with a static condition can only be marked pending during a checkpoint if the condition is true at the beginning of that checkpoint.
 - c. A conditional ability with a static condition can only be marked pending if no other instance of the same ability from the same source is already pending, imminent, or resolving.
 - d. A conditional ability with a static condition can only be marked pending if the instructions in the ability have the potential to change the game state.

Example: Parasite is hosted on a piece of ice with 0 strength. Parasite's ability tries to trash that ice. If the effect of an interrupt ability prevents the ice from being trashed without increasing its strength, the condition is still true, so the ability will become

pending again. But this does not occur while the previous instance of the ability is still pending or resolving. Conversely, if a static ability (such as the one on Architect) prohibits Parasite from trashing the ice, Parasite's ability does not have the potential to change the game state, and therefore it does not become pending.

- 9.6.8. A player can trigger a conditional ability while they have priority in a reaction window in which the ability is pending.
- a. Once a player triggers a conditional ability, that instance of the ability loses its pending status. Other pending instances of the ability are unaffected and can still be triggered later, regardless of whether those instances are associated with the same reaction window.
 - b. If the [interrupt] symbol appears before a conditional ability's text, that ability is an interrupt. These abilities will become pending in interrupt windows rather than reaction windows, but are otherwise resolved in the same way as non-interrupt conditional abilities. See [section 9.9](#).
- 9.6.9. If a conditional ability gives its controller a choice of whether to apply its effects, such that the ability could potentially have no effects at all, it is considered an OPTIONAL CONDITIONAL ABILITY. These abilities can usually be recognized by the presence of permissive words such as "may" or "allows", or by a restriction such as "Use this ability only once per turn." If the ability is not optional, then it is a MANDATORY CONDITIONAL ABILITY.
- a. Instances of optional conditional abilities can still be pending when their controller passes in the corresponding priority window. Players are not required to trigger optional conditional abilities.
 - b. Players must trigger all instances of pending mandatory conditional abilities they control in a given priority window before they can pass in that window. See [section 9.2.8](#).
 - c. Both optional and mandatory conditional abilities may have optional parts to their effects. Even if triggering the ability is mandatory, its controller may still decline any optional constituent effects that arise during its resolution.
 - d. A conditional ability and its source are considered used when any optional component of the ability's effects is carried out.
- 9.6.10.** If any instances of a conditional ability are pending and the conditional ability itself becomes inactive, those instances lose their pending status and will not resolve.

Example: The Runner has both Aesop's Pawnshop and Drug Dealer installed. Both cards have abilities that meet their trigger conditions when the Runner's turn begins. If the Runner chooses to trigger Aesop's Pawnshop first, and then uses it to trash Drug Dealer, Drug Dealer's pending ability can never be triggered, so the Runner does not lose a credit.

- 9.6.11.** If a priority window closes while any abilities still have pending instances associated with that window, the remaining instances lose their pending status and cannot be triggered or resolved. See [rule 9.2.8f](#).
- 9.6.12.** Once a player triggers a conditional ability and its instructions become imminent, the ability becomes independent of its source and [rule 9.1.4](#) applies to it.
- 9.6.13.** A DELAYED CONDITIONAL ABILITY is a conditional ability maintained by a lingering effect (see [section 9.10](#)). Unlike other lingering effects, some delayed conditional abilities are created without an explicitly stated duration.
- a.** If an “if successful” ability contains a nested delayed conditional ability referring to breaching the attacked server, that ability applies to the breach at [step 6.9.5b](#) of the run the “if successful” ability is associated with. Its duration expires at the end of the run. See [rule 6.7.4b](#).
 - b.** If an instruction creates a delayed conditional ability and specifies a duration, **the delayed conditional ability is active until that duration expires.**

Example: In the Groove creates a delayed conditional ability that specifies a duration of “this turn”. Since a duration is specified, it is not limited to only being triggered once. The ability becomes pending and can resolve every time its trigger condition is met, and its existence only expires at the end of the turn.

- c.** **Any other delayed conditional ability exists until the next time it resolves, after which it expires.**

Example: Joshua B’s ability allows the Runner to gain [click]. If they do, it creates a delayed conditional ability that is independent of Joshua B itself, which will meet its trigger condition only once, when that turn ends. After that ability resolves, the game no longer maintains the lingering effect creating the ability.

9.6.14. Steps of Triggering and Resolving a Conditional Ability

- a. Announce that you will trigger one of the pending abilities associated with the reaction window that gave you priority.
- b. Announce any targets for the ability’s first instruction. That instruction then becomes imminent.
- c. An interrupt window occurs, during which abilities can modify, prevent, or avoid the imminent effects.
- d. Resolve the instruction, applying any changes to its effects from interrupts that were resolved. If the controller of the ability chooses to resolve an optional effect contained in this instruction, and this is the first instruction for which that player has done so, “when used” conditionals meet their trigger conditions.
- e. A checkpoint occurs.

- f. If there are more instructions to resolve, announce any targets for the next instruction. That instruction then becomes imminent. Return to (c).
- g. Resolution of the conditional ability is complete.

9.7. Play Abilities

- 9.7.1. Any ability on an event or operation that is not a paid, conditional, or static ability is a **PLAY ABILITY**. Play abilities are the abilities that resolve as part of playing that event or operation during [step 8.6.6f](#).

*Example: The operation card Hyoubu Precog Manifold has several abilities of different types. The first sentence reads “Play only if there is no active **lockdown**.” This is a static ability that contains a restriction and no declarations. The second sentence (“This operation is not trashed until your next turn begins.”) is a play ability that has no immediate impact on the game state but creates a lingering effect that alters how the operation will leave the play area after resolving. The second paragraph reads simply, “Choose a server.” This play ability instruction is the main part of the card that actively affects the game state while the operation resolves. Finally, the third paragraph is a single conditional ability with multiple instructions that is active as long as the operation remains in the play area.*

9.7.2. Steps of Resolving a Play Ability

- a. Announce any targets for the ability’s first instruction. That instruction then becomes imminent.
- b. An interrupt window occurs, during which abilities can modify, prevent, or avoid the imminent effects.
- c. Resolve the instruction, applying any changes to its effects from interrupts that were resolved.
- d. A checkpoint occurs.
- e. If there are more instructions to resolve, announce any targets for the next instruction. That instruction then becomes imminent. Return to (b).
- f. Resolution of the play ability is complete.

9.8. Subroutines

9.8.1. **A SUBROUTINE is an ability beginning with the [sub] symbol. Only ice can have subroutines.**

9.8.2. **Subroutines are the only abilities that are ordered.**

- a. **The order of the subroutines printed on a piece of ice is the order in which they appear on that ice.**

- b. Unless otherwise stated, subroutines a piece of ice gains from an ability are added after all other subroutines the ice has at that time.

Example: The Runner encounters Marker, and its subroutine resolves. The next ice the Runner encounters is Brainstorm. When the encounter begins, the lingering effect from Marker gives Brainstorm an “[sub] End the run.” subroutine. Then, Brainstorm’s conditional ability resolves, giving it the appropriate number of “[sub] Do 1 brain damage.” subroutines. The “[sub] Do 1 brain damage.” subroutines are all ordered after the “[sub] End the run.” subroutine.

- c. If multiple subroutines are added to a piece of ice at the same time in an unspecified order, the Corp declares the relative order of those subroutines at the time the ice gains them.

9.8.3. Some ice have static abilities that grant that piece of ice a variable number of subroutines. Subroutines from such an ability are always ordered before any other subroutines on that ice.

- a. If the number of subroutines granted to a piece of ice by its own static ability increases, the additional subroutines are added immediately after the previously-existing subroutines from the same ability.

Example: The Corp has no bad publicity and controls a rezzed copy of Ileress hosting a condition counter from Sub Boost. If the Corp takes bad publicity, the corresponding subroutines from Ileress’s ability are added before the “[sub] End the run.” from the condition counter.

Example: The Corp has 3 cards in HQ while the Runner is encountering Ashigaru. After the Runner breaks Ashigaru’s 3 subroutines, the Corp uses Panic Button to draw another card. The new subroutine is added after the previous 3, so the Runner can use Gbahali to break it.

- b. If the number of subroutines granted to a piece of ice by its own static ability decreases, subroutines are lost beginning with the last subroutine the ice previously had from the same ability and working backwards.

Example: The Corp has 3 cards in HQ while the Runner is encountering Ashigaru. The Runner breaks the first subroutine, then uses Utopia Shard to force the Corp to discard 2 cards. Ashigaru correspondingly loses its last 2 subroutines, leaving behind the subroutine that was already broken.

9.8.4. One card, *Hive*, has a static ability that removes its own printed subroutines. Subroutines are lost to this ability beginning with the last printed subroutine and working backwards.

9.8.5. For each encounter, the subroutines on the encountered ice have a status of **BROKEN** or **UNBROKEN** with respect to that encounter.

- a. When an encounter begins, each of its subroutines becomes unbroken with respect to that encounter.
 - b. When a piece of ice gains 1 or more new subroutines during an encounter with it, those subroutines come into existence unbroken with respect to that encounter.
 - c. When an encounter ends, card abilities can still check whether subroutines were broken or unbroken during that encounter.
- 9.8.6. To BREAK a subroutine is to change its status with respect to the active encounter from unbroken to broken. Many cards that can allow a player to break subroutines are icebreakers (see [section 3.9.5](#)). If the Runner breaks all the subroutines on the encountered ice, they fully break that ice (see [section 6.5.7](#)).
- 9.8.7. Only unbroken subroutines can be chosen as targets for abilities that would break them.
- a. An ability that breaks “all subroutines” does not target any subroutines, and can be used as long as the encountered ice has at least 1 unbroken subroutine.
 - b. One card, *Grappling Hook*, has the ability to break “all but 1 subroutine” on the encountered ice. To trigger this ability, its controller chooses as its target the 1 subroutine that will not be broken. When the ability resolves, it breaks all unbroken subroutines except the chosen subroutine.
- Example: The Runner uses two copies of Grappling Hook to break all of the subroutines on Heimdall 1.0. For the first Grappling Hook, the Runner chooses the unbroken “[sub] Do 1 brain damage.” subroutine as the target, and Grappling Hook breaks both “[sub] End the run.” subroutines. For the second Grappling Hook, the Runner chooses one of the already-broken “[sub] End the run.” subroutines as the target. They can choose the broken subroutine as Grappling Hook’s target because Grappling Hook will not attempt to break the chosen subroutine. When the ability resolves, Grappling Hook breaks the “[sub] Do 1 brain damage.” subroutine, but does nothing to the other “[sub] End the run.” subroutine, because it is already broken.*
- 9.8.8. At [step 6.9.3c](#) of an encounter, the Corp resolves subroutines that are unbroken with respect to that encounter.
- a. Resolving subroutines is mandatory and does not open a priority window.
 - b. The Corp resolves the unbroken subroutines 1 at a time in order. The order of subroutines is discussed in [section 9.8.2](#).
 - c. If the encounter ends during this process, no more subroutines are resolved.

Example: The Runner encounters Little Engine and does not break any subroutines. Little Engine's first subroutine resolves and ends the run. Since the run and therefore the encounter have ended, no more subroutines resolve. The Runner will not gain 5[c].

- d. One card, *Street Magic*, allows the Runner to set the order in which subroutines are resolved. If this ability is active when [step 6.9.3c](#) is reached for the first time in an encounter, the Runner declares an order for the unbroken subroutines, and the Corp must resolve them in this order instead of their normal order. The new order is determined before any subroutines are resolved.

- 9.8.9. Once a subroutine's first instruction becomes imminent, the subroutine becomes independent of its source and [rule 9.1.4](#) applies to it.

9.8.10. Steps of Resolving a Subroutine

- a. The subroutine itself becomes imminent.
- b. An interrupt window occurs, during which abilities can prevent the subroutine from resolving. If this happens, the remaining steps are not carried out.
- c. Announce any targets for the subroutine's first instruction. That instruction then becomes imminent.
- d. An interrupt window occurs, during which abilities can modify, prevent, or avoid the imminent effects.
- e. Resolve the instruction, applying any changes to its effects from interrupts that were resolved.
- f. A checkpoint occurs.
- g. If there are more instructions to resolve, announce any targets for the next instruction. That instruction then becomes imminent. Return to (d).
- h. Resolution of the subroutine is complete.

9.9. Interrupts and Replacement Effects

- 9.9.1. An INTERRUPT is an ability that modifies either the effects of another instruction or the context in which that instruction will be resolved. Interrupts can be written in the form of a paid ability or a conditional ability.
 - a. Interrupt abilities are denoted by the [interrupt] symbol appearing at the beginning of the ability, or by certain keywords in the ability's text.
 - b. Any paid or conditional ability that uses the word "prevent", "avoid", or "**would**" is an interrupt ability.

- 9.9.2. The EXPECTED EFFECTS of an imminent instruction are the effects described by the instruction’s text, modified by any static abilities that affect it and any replacement effects or interrupt effects that have been applied. The expected effects of an instruction are continually updated throughout the interrupt window.

Example: The Runner plays Process Automation and the instruction “Gain 2[c] and draw 1 card.” becomes imminent. The expected effects of this instruction are that the Runner will gain 2[c] and draw 1 card.

Example: The Runner plays Process Automation after Lockdown’s subroutine resolves. The instruction “Gain 2[c] and draw 1 card.” becomes imminent, but the Runner cannot draw cards. The expected effect of this instruction is that the Runner will gain 2[c].

- a. When an instruction resolves, its expected effects correspond to the effects that occur, except for situations where [rule 9.9.7d](#) applies.

- 9.9.3. In order to trigger an interrupt, it must be RELEVANT to an imminent instruction. An interrupt is relevant to an instruction if at least one of the following is true:

- a. Its effects could prevent or avoid part or all of the imminent instruction’s expected effects. See [section 9.9.5](#).
- b. Its effects could modify a value associated with the imminent instruction’s expected effects. See [section 9.9.6](#) and [section 9.9.7](#).
- c. Its effects could create a replacement effect that applies to the imminent instruction’s expected effects. See [rule 9.9.10](#).
- d. It has a trigger condition using the word “would” that is met by the imminent instruction’s expected effects.

Example: The trigger condition on The Class Act’s second ability is “The first time each turn you would draw any number of cards”. Even though this ability’s effects do not directly modify the imminent instruction’s effects, the ability is relevant to an instruction that is expected to draw cards.

- 9.9.4. Whenever an instruction becomes imminent, an interrupt window opens to allow players to modify the effects of that instruction. [Section 9.2.9](#) contains the rules for handling priority during interrupt windows.

- a. As the window opens, the initial expected effects of the instruction are determined, then any active replacement effects that replace all or part of the expected effects are applied. If multiple replacement effects could apply, the order in which they are applied follows the rules in [section 9.9.11](#).
- b. After replacement effects have been applied, but before players receive priority, active conditional ability interrupts that are relevant to the imminent instruction are marked pending and are associated with the window. If a conditional ability interrupt becomes active after the interrupt window opens, it does not become pending.

- c. While a player has priority in an interrupt window, they can trigger a conditional ability interrupt they control only if it is pending in that window and still relevant to the imminent instruction.

Example: An instruction is imminent that would trash an installed copy of Harbinger. The Runner uses Sacrificial Construct to prevent Harbinger from being trashed. The Runner can no longer trigger Harbinger's interrupt ability, even though it is still pending, because the expected effects of the instruction no longer include Harbinger being trashed and therefore the ability is no longer relevant.

Example: An instruction is imminent that would give the Runner 2 tags. The Runner trashes Decoy to use its ability to avoid 1 of the tags, which meets the trigger condition for Thunder Art Gallery's ability. The latter resolves as a chain reaction (see [rule 9.1.2a](#)) while the interrupt window is still open, allowing the Runner to install No One Home, which has a conditional ability interrupt that is relevant since the instruction is still expected to give the Runner 1 tag. However, No One Home was not active when the interrupt window opened, so its ability is not pending and cannot be triggered.

- d. While a player has priority in an interrupt window, they can use a paid ability interrupt that is relevant to the imminent instruction. The ability does not have to have been active when the interrupt window opened, nor does a paid ability window need to be open at this time.

Example: An instruction is imminent that would give the Runner 2 tags. The Runner trashes Decoy to use its ability to avoid 1 of the tags, which meets the trigger condition for Thunder Art Gallery's ability. The latter resolves as a chain reaction (see [rule 9.1.2a](#)) while the interrupt window is still open, allowing the Runner to install another copy of Decoy. The Runner receives priority again in the original interrupt window and uses the second Decoy's ability to avoid the other imminent tag.

9.9.5. To PREVENT or AVOID an effect is to stop that effect from happening. The terms “prevent” and “avoid” are synonymous.

- a. Some abilities look for an ordinal instance of an effect that “would” take place within a specified time period. These abilities look for the appropriate number of times that effect becomes imminent, not whether the effect actually resolves. Preventing or replacing an effect does not allow these abilities to meet their trigger conditions again in the same time period.

Example: Tori Hanzō's interrupt can only be used “the first time you would do net damage” during a run. If the first net damage that becomes imminent is prevented by the Runner using Feedback Filter, Tori Hanzō cannot be used on any other instances of net damage for the remainder of the run, as the next imminent net damage is the second time the Corp “would” deal net damage, not the first time.

- b. Abilities that do not use “would” do not see effects that are prevented, as those effects did not occur.

- c. A static ability can stipulate that a particular effect is always prevented. This kind of prevention effect does not require a player to explicitly trigger it.

Example: Paparazzi reads in part, “Prevent all meat damage.” This is a static ability that applies automatically to any instruction that would deal meat damage.

9.9.6. Some instructions make use of an associated value. These values can be changed by interrupts and replacement effects.

- a. A number of tags the Runner would take from an effect is a value. Avoiding a number of those tags decreases that value. The value for a number of tags the Runner takes must be greater than 0 for the effect giving tags to occur.
- b. An amount of damage the Runner would take from an effect is a value. Preventing an amount of that damage decreases that value. The value for an amount of damage the Runner takes must be greater than 0 for the damage effect to occur.
- c. A cost that would be paid while resolving an effect is a value. [Rule 1.16.1c](#) applies to the final value at the time the cost is paid. See also [rule 1.16.1d](#).

Example: Patchwork’s interrupt modifies a play cost or install cost, so it is relevant to any instruction where a card will be played or installed and the corresponding cost paid.

- d. The base trace strength of a trace attempt is a value. See [section 10.8](#) for rules about trace attempts. A base trace strength value does not need to be greater than 0 for the trace attempt to occur.

9.9.7. Values associated with an imminent instruction are calculated continuously as part of the instruction’s expected effects.

- a. While an instruction is imminent, values associated with that instruction can be reduced below 0. Values 0 or lower can still be modified by other interrupt abilities while the instruction remains imminent.

Example: The Runner takes a tag on their turn, allowing the Corp to trigger Mr. Stone. While the 1 meat damage from Mr. Stone’s ability is imminent, both players can trigger interrupts to modify the damage effect. Since it is the Runner’s turn, they receive priority first and use Biometric Spoofing to prevent 2 damage. The imminent effect is now expected to do -1 damage. When the Corp receives priority, they trigger two copies of The Cleaners that each add 1 to the expected damage, increasing the total to 0 and then to 1. The ability will resolve with 1 damage even though the amount of damage was temporarily 0 or less while the effect was imminent.

- b. If an effect prevents “all damage” or “any amount of damage” while an instruction is imminent, the damage is removed from the expected effects entirely, and there is no longer a value to be modified. If an effect avoids “all tags” or “any number of tags” while an instruction is imminent, taking tags is removed from the expected effects entirely, and there is no longer a value to be modified.

- c. One card, *Bio-Modeled Network*, has an instruction that prevents “all but 1 net damage.” This ability is relevant when the imminent instruction’s expected effects include doing net damage and the amount of the damage is 2 or more. The effect of this ability is to set the value for the amount of damage to 1. Previously-applied modifiers to the value are ignored. The value can still be modified by other interrupts after being set to 1 this way.
- d. Some types of values must be greater than 0 for the associated effect to occur. See [section 9.9.6](#). At the time an instruction resolves, if any value of one of those types is 0 or less, as much of that ability resolves as possible without applying the part of the effect making use of that value, following rules [1.2.3](#) and [1.2.4](#) of the Golden Rules.

Example: The Runner accesses Breached Dome. While the instruction from its “when accessed” ability is imminent, the Runner uses Biometric Spoofing to prevent 2 damage. When the ability resolves, the value for the amount of damage is -1, which is less than or equal to 0, so the damage effect cannot occur and the Corp only trashes the top card of the stack.

- e. Any effects making use of modified values maintain their restrictions and specifications with the new values.

Example: While the result of a successful trace from Flare is imminent, the Corp triggers The Cleaners to increase the imminent meat damage by 1. Even though the damage from Flare has been increased from 2 to 3, none of that meat damage can be prevented.

9.9.8. A REPLACEMENT EFFECT modifies another effect by replacing some or all of the original effect’s behavior with different behavior. Replacement effects are indicated by the word “instead” in their text. Once created, replacement effects apply automatically and do not require players to explicitly trigger them.

- a. Interrupts can introduce replacement effects that apply to the currently imminent instruction. See [rule 9.9.10](#).
- b. Static abilities can stipulate replacement effects that apply whenever appropriate while the static ability is active.
- c. Other abilities can create replacement effects ahead of time, which apply to effects that could happen later. This is a type of lingering effect (see [section 9.10](#)).

9.9.9. Replacement effects most commonly apply to the effects of an instruction, but some apply to static abilities or alter the durations of lingering effects.

- a. If a replacement effect modifies the effects of a static ability or the duration of a lingering effect, this modification applies continuously for as long as the replacement effect itself is active, and no longer applies once the replacement effect or the original effect being modified becomes inactive.

Example: Gebrselassie's static ability replaces the durations of certain lingering effects affecting its host with a duration of "for the remainder of the turn." If Gebrselassie is uninstalled or moved to a different host, its replacement effect no longer applies to those lingering effects, so they revert to their normal durations. If any of those durations have already expired, the corresponding effects will end at the next checkpoint.

- b. If an active replacement effect can apply to the expected effects of an imminent instruction, it is applied as the interrupt window opens, and the expected effects of the instruction are updated accordingly. This happens before the expected effects are checked for relevant active abilities that could become pending. Replacement effects created by interrupts will not yet be active at this time and are applied according to [rule 9.9.10](#) instead.
- c. Each replacement effect can only apply once to any single effect. Once a replacement effect has been applied, even if the new effect it creates (or a new effect created by a later replacement effect) still includes an occurrence of the effect that gets replaced, the same replacement effect cannot be applied to that effect again.

Example: The Runner accesses Project Vacheron and its interrupt ability resolves. The interrupt creates a replacement effect that will override adding the agenda to the Runner's score area. The Runner will still steal Project Vacheron, but they will do so by adding it to their score area with hosted agenda counters. Even though the modified effect still includes adding Project Vacheron to the Runner's score area, there is no way for the replacement effect to apply again to its own results.

- 9.9.10. Some interrupts create replacement effects to modify an instruction that is already imminent. When the interrupt resolves, if the replacement effect can apply to the expected effects of the imminent instruction, it is applied immediately.

Example: An instruction is imminent with an expected effect of the Corp doing net damage to the Runner. The Corp uses Tori Hanzō's ability, creating a replacement effect replacing net damage with brain damage. The replacement effect applies immediately, so after Tori Hanzō's ability resolves, the expected effect is now to do 1 brain damage. When players get priority again, their interrupts are relevant or not based on the new expected effects.

- 9.9.11. If multiple replacement effects would apply to an effect acting on a targeted card, that card's controller chooses the order in which the replacement effects apply. If multiple replacement effects would apply to any other effect, the controller of the base effect chooses the order in which the replacement effects apply.
- a. After applying one replacement effect, a subsequent replacement effect can only be applied if the effect it was to replace is still expected. A replacement effect cannot apply without something to replace.

Example: The Runner chooses HQ with Security Testing when their turn begins, then plays Account Siphon. When the run becomes successful, both Security Testing and Account Siphon create replacement effects. When [step 6.9.5b](#) of the run becomes

imminent, the Runner must choose to apply either the replacement effect from Security Testing or the one from Account Siphon. Since neither effect creates a new instance of accessing cards, the effect not chosen to apply first has nothing to replace and does not apply.

Example: The Runner chooses R&D with Security Testing when their turn begins, then plays Showing Off. When a run on R&D becomes successful, both Security Testing and Showing Off create replacement effects. When [step 6.9.5b](#) of the run becomes imminent, the Runner must choose to apply either the replacement effect from Security Testing or the one from Showing Off. If they choose to apply Security Testing first, Showing Off's effect has nothing to replace and does not apply. If they choose to apply Showing Off first, the expected effects still include accessing cards from R&D, albeit not in the usual way. Security Testing can still replace this access, so it applies next. Either way, the Runner will gain 2[c] and will not access any cards.

9.10. Lingering Effects

- 9.10.1. Instructions can create LINGERING EFFECTS that exist beyond their resolution. Once a lingering effect is created, it exists independently of its source even if the source becomes inactive. Lingering effects **remain active** only for the duration specified in the instruction that created them. **Lingering effects with durations that have expired are removed from the game state at [step 10.3.1b](#) of each checkpoint.**
- 9.10.2. Some lingering effects exist only to maintain the existence of another ability. Delayed conditional abilities are maintained this way and **may have an implicit duration**. See [section 9.6.13](#).
- 9.10.3. **Some lingering effects maintain a choice that a player made while resolving an ability in order for another ability or effect with the same source object to make use of that choice.**
 - a. **If the choice is only referred to by another lingering effect, the lingering effect maintaining the choice has the same duration as the other lingering effect.**

Example: The Runner resolves Pelangi's paid ability and chooses an ice subtype. The ability then creates a lingering effect adding the chosen subtype to the encountered ice. Both the choice of subtype and the effect adding it to the ice expire at the end of the encounter.

- b. **If the abilities linked in this way use the word "this" in reference to a timing structure, then the reference applies only to the occurrence of that timing structure in which the choice was made. In this case, the duration of the lingering effect maintaining the choice expires when that timing structure ends or the source object becomes inactive.**

Example: Security Testing's first ability allows the Runner to choose a server. Its second ability has the trigger condition, "the first time this turn you make a successful run on the chosen server". The reference to "this turn" in this

condition indicates that the choice should be maintained only until the timing structure of the Runner's turn ends. On future turns, the previous choice of server will not affect when or whether Security Testing's second ability meets its trigger condition, even if a new server is not chosen.

- c. If the previous cases do not apply, the duration of the lingering effect expires only when the source object becomes inactive.

Example: Femme Fatale's first conditional ability directs the Runner to choose an installed piece of ice. Which ice was chosen is maintained as a lingering effect for as long as Femme Fatale remains active, and its last ability refers to the object that is being remembered by that lingering effect.

- 9.10.4. If a lingering effect is created with a duration based on a timing structure that is not in progress at the time the lingering effect is created, the duration expires immediately, and the lingering effect persists only until the next checkpoint after it is created.

- a. Abilities on an **icebreaker** that modify their source's strength have an implicit duration of "for the remainder of the current encounter". See [rule 3.9.5b](#).

- 9.10.5. **Only lingering effects have durations. An ability that modifies the durations of other abilities functions by keeping their corresponding lingering effects active until an additional duration expires. This type of ability does not interact with effects from static abilities.**

Example: The Runner controls Gebrselassie hosted on Na'Not'K. They run a server protected by 3 pieces of ice, and use Na'Not'K's last ability to raise its strength to a total of 6. When the run ends, Na'Not'K's static ability will no longer provide +3 strength, but the lingering effect of +2 strength from its paid ability will be maintained by Gebrselassie until the turn ends or Gebrselassie stops being hosted on Na'Not'K. If the Runner makes another run on a server protected by only 1 piece of ice, Na'Not'K will have 4 strength when that run begins.

9.11. Identifying Instructions

To correctly resolve conditional abilities and interrupts that could meet their trigger conditions during the resolution of another ability (see [rule 9.1.2a](#)), players need to determine where checkpoints occur during the resolution of each ability. This section aids players in making this determination. For clarifications on particular cards, refer to the Release Notes for the set in which that card appears or to the Rulings section of that card's page on netrunnerdb.com.

- 9.11.1. An instruction normally cannot be paused once it begins to resolve. An instruction can only be paused if a checkpoint opens in the middle of its resolution, which only occurs in the following cases:
 - a. Any time a cost is paid, a checkpoint must occur immediately afterward.
 - b. When a timing structure is opened, the checkpoints in that timing structure are carried out. See [section 9.2.2](#) for a list of timing structures.

- c. When a player draws cards, a checkpoint occurs at [step 8.4.5b](#) of that procedure.
 - d. When an event or operation is played, a checkpoint occurs at [step 8.6.6e](#) of that procedure.
 - e. When a trace is initiated, a checkpoint occurs at [step 10.8.6b](#) of that procedure.
- 9.11.2. Each step in a timing structure forms a single instruction, and is accordingly preceded by an interrupt window and followed by a checkpoint.
- a. Some game procedures that are not timing structures are presented as a sequence of steps. These steps do not indicate instruction boundaries, and checkpoints only occur in these procedures when explicitly called for.

Example: The steps of installing a card are not separate instructions. Installing a card usually happens in its entirety as part of a single instruction. The only checkpoint that occurs during the procedure of installing a card is at [step 8.5.13d](#), immediately after the install cost is paid.

Example: The steps of a checkpoint are not instructions. No checkpoints take place in the middle of another checkpoint.

- 9.11.3. Usually, each sentence in the text of an ability forms a single instruction. After each instruction, an ability pauses its resolution to allow priority windows to open and other abilities to resolve. First, a checkpoint occurs, allowing any appropriate conditional abilities to be marked as pending in a reaction window, then targets are announced for the next instruction. Finally, the next instruction becomes imminent, allowing interrupts relevant to that instruction to resolve in an interrupt window.
- 9.11.4. Some sentences are not instructions, some sentences encompass multiple instructions, and some instructions encompass multiple sentences. These are the cases in which sentences do not correspond one-to-one with instructions:
- a. A sentence is not part of any instruction if it only provides clarification, restrictions or conditions on when or how the ability it appears in can be triggered, or otherwise does not give directions to the players or the game state that would be carried out as the ability resolves.

*Example: A sentence reading "Use this ability only by spending credits from a **stealth** card." is not an instruction or part of any instruction. It is a restriction that applies to the entire ability.*

- b. If a sentence directs a player to play, install, **or access** more than one card, each such card is handled as a separate instruction. The instruction ends and a new one begins immediately before each play, install, **or access** after the first.

Example: The Corp plays Shipment from MirrorMorph, which has a single sentence directing them to install up to 3 cards from HQ. This is treated as though it said, "You

may install a card from HQ. You may install a card from HQ. You may install a card from HQ.”, or equivalently, “You may install a card from HQ. Repeat this process 2 more times.”

- c. If a sentence directs a player to choose 1 or more targets, but does not act on that choice or describe any other effects, that sentence and the following sentence form a single instruction.

*Example: Tinkering’s play ability reads, “Choose a piece of ice. That ice gains **sentry**, **code gate**, and **barrier** until the end of the turn.” The first sentence does nothing except direct the Runner to select a target, so it forms a single instruction with the second sentence. The target is chosen as the instruction becomes imminent, and the card gains the subtypes when the instruction resolves.*

- d. Some older cards have search effects written in the same sentence as the effects that will be performed upon any found cards. Treat these sentences as if ending the search and performing any necessary shuffling is the end of an instruction. A checkpoint occurs while cards found by the search are in the set-aside zone, then the remainder of the sentence is treated as the next instruction.

Example: The Runner uses Djinn to search for a program while the Corp has Personality Profiles in their score area. The Runner finds and sets aside a copy of Datasucker and shuffles the stack. This ends an instruction, and since the search has occurred, Personality Profiles’s ability meets its trigger condition. The Corp resolves the ability during a reaction window before the next instruction becomes imminent, forcing the Runner to trash a random card from their grip. Finally, the second instruction resolves and the Runner adds the set-aside Datasucker to their grip.

- e. Some older cards direct a player to look at or reveal a set of cards in the same sentence as the effects that will be performed upon those cards. Treat these sentences as if making the cards visible to the relevant player(s) is the end of an instruction. A checkpoint occurs once the cards are visible to the relevant player(s), then the remainder of the sentence is treated as the next instruction.

Example: The Corp is resolving the first subroutine on Architect, which, in one sentence, allows the Corp to look at cards in R&D and install one of them. Looking at the top 5 cards of R&D ends the ability’s first instruction. Once the Corp is able to see those cards, they can choose a target for the second instruction, which will install it. Since the install is optional, the Corp can also decline to choose a target, and the second instruction will have no effect. After the second instruction resolves, the ability is complete, and the Corp’s permission to look at cards in R&D ends.

- f. If an effect directs a player to choose between a set of options that would create different effects, that choice ends an instruction. Each option begins its own instruction or set of instructions, and the one chosen will resolve next.

Example: The Runner encounters Data Raven and is forced by its first ability to resolve either “take 1 tag” or “end the run.” Making this choice ends the first instruction, and the option the Runner chooses will become imminent after a checkpoint.

- 9.11.5. If an ability initiates a timing structure, its source may have linked abilities that apply during that timing structure. These are frequently written directly after the instruction that creates the timing structure, but they are not instructions in the same ability.

9.12. Other Rules and Terminology

9.12.1. Simultaneous Effects

- a. If more than one effect attempts to modify the same value, determine its final value from its default value by first applying any effect that sets it to a specific value, then applying each effect that increases the value, and finally applying each effect that lowers the value.
- b. If more than one effect attempts to add or remove the same subtype from a card, count the number of times the subtype is added (including from the card's printed text) and the number of times the subtype is removed. As long as the number of times added is greater, the card has that subtype. As long as the number of times removed is greater or the numbers are equal, the card does not have that subtype.
- c. If more than one effect gives a player a choice of how to resolve an ability or game rule such that both players are instructed to make a choice that can only be made once, the active player makes that choice. The effects that granted the players the choice otherwise resolve as normal.

Example: If Chronos Protocol and Titanium Ribs are both active, both players are instructed to choose the card trashed by the first net damage each turn. Since this is impossible, only the active player chooses that card. Since Chronos Protocol also instructs the Corp to look at the Runner's grip and this is not prevented or contradicted by any other ability, this part of the effect happens on either player's turn.

9.12.2. Acting on Sets of Cards

- a. If an effect acts on a set of cards for a single purpose, then the effect acts on all of those cards together simultaneously rather than one at a time.

Example: The Runner uses Singularity to simultaneously trash a rezzed Warroid Tracker and a rezzed Hostile Infrastructure. Hostile Infrastructure has a trigger condition of "Whenever the Runner trashes a Corp card", so it sees both trashed cards and becomes pending twice. By contrast, Warroid Tracker has a trigger condition of "Whenever the Runner trashes 1 or more cards in or protecting this server," so it sees only a single event that trashed multiple cards. Warroid Tracker's ability becomes pending only once.

- b. When the resolution of an ability involves an effect repeated "for each" of some quantity, the full effect of the ability is first calculated, and then the ability resolves. If the ability would resolve on an empty set or a value of 0, then nothing is counted and thus the effect does not resolve.

Example: The Runner encounters a Cortex Lock and does not break the subroutine. If the Runner has 2 unused MU, both of the net damage can be prevented with Biometric Spoofing because Cortex Lock does a single instance of 2 net damage, not 2 instances of 1 net damage each. If the Runner has no unused MU, then no net damage is dealt because there is no MU to count.

- c. If a condition refers to “all” items in a set and that set contains zero items, the condition is automatically satisfied as soon as it would be satisfied for one or more of that item.

Example: The Runner plays Forked, initiating a run on a server. The first piece of ice that the Runner encounters is Troll. If Troll’s “when encountered” ability does not end the run, the Runner is automatically considered to have broken all of the zero subroutines on Troll as soon as [step 6.9.3b](#) of the encounter begins, and Troll is trashed by Forked.

9.12.3. “Must”

- a. If a card ability states that a player “must” do something without stipulating how that player must do so, then that player is forced to make any decisions necessary to satisfy the requirement, even if it requires the use of other card abilities.

Example: The Runner has an Imp and a Scrubber installed and accesses Mumbad Virtual Tour. If the Runner only has 4[c], they must still choose to either spend a counter from Imp to trash Mumbad Virtual Tour if they are able to do so or spend recurring credits on Scrubber to help pay the trash cost of the Mumbad Virtual Tour.

- b. If a card ability states that a player “must” do something and specifies the means by which that player must do so, then that player is not required to carry out the effect except by the means specified, even if it would be possible through the use of other card abilities.

Example: The Runner has Imp and Neutralize All Threats installed when they access a card with a trash cost greater than the number of credits they have available to spend. Because they are unable to trash the card by paying its trash cost, they are not required to spend a counter from Imp in order to trash it.

- c. If the “must” ability presents a player with a choice between two or more effects, that player must choose an effect that can be fully resolved. If none of the choices can be fully resolved, then the “must” ability does nothing.

Example: If the first subroutine on Fairchild 2.0 resolves while the Runner has no installed cards but 3[c], then the Runner must choose to pay 2[c]. When the second subroutine on Fairchild 2.0 resolves, the Runner can neither pay 2[c] nor trash an installed card, so nothing happens.

- d. Making a choice between effects is always a separate instruction from resolving the chosen effect (see [rule 9.11.4f](#)). The previous rule only requires the player to choose an effect that is possible to resolve; it does not require the chosen effect to

actually resolve as expected. Once an effect has been chosen, the player can apply interrupts to modify, prevent, or replace that effect as normal.

Example: If the Runner encounters Data Raven, they must choose to either take a tag or end the run. The Runner can choose to take the tag, then avoid the tag with Decoy. This does not cause the run to end, as the decision to take the tag has already been made.

- e. A singular “must” ability cannot force a player to pay an additional cost they wish to decline, but a player cannot avoid choosing a fully resolvable effect from among multiple options in a “must” ability by declining to pay the additional cost for one of those choices.

Example: If Service Outage is active and the Runner has Always Be Running installed, the Runner can decline to pay the additional 1[c] to make a run with their first [click], thus satisfying the “must” of Always Be Running and allowing them to spend their first click on a different action. However, if a Runner plays a Forged Activation Orders on an unrezzed Archer, the Corporation cannot choose to rez the Archer but decline to forfeit an agenda; the Corp must either pay 4[c] and forfeit an agenda to rez the Archer or trash the Archer.

9.12.4. “Repeat this process”

- a. If an ability instructs a player to “repeat this process” without specifying how many times to do so, then that player resolves the ability again, including the instruction to “repeat this process” if it is still applicable.
- b. If an ability instructs a player to “repeat this process” a specified number of times, then that player resolves the entire ability, except for the repetition instruction, that many times in a row. The number of repetitions is determined after the first full resolution but before the first repetition, and once the number of repetitions is determined it cannot be changed.
- c. Regardless of the number of times an ability is repeated, a checkpoint occurs after each full resolution of the ability’s instructions, before another repetition begins.
- d. Each repetition of the ability is resolved independently of the others. If the ability provides the player with any choices, they may choose differently each time.

9.12.5. “Persistent”

- a. **A PERSISTENT ability is an ability on a Corp card marked with the “persistent” ability flag before its other text. When the Runner trashes a rezzed card they are accessing, its persistent abilities PERSIST through a lingering effect that is created as the source card is trashed. See also [section 9.10](#).**
- b. **A persistent ability begins to persist simultaneously with the trashing of its source card. The ability never becomes inactive, nor is it considered a new ability.**

- c. An ability only persists until just after the end of the run during which it began to persist. When the reaction window immediately following [step 6.9.6d](#) of that run closes, the lingering effect allowing the ability to persist expires, and the ability ceases to persist.
- d. A persisting ability is only applicable to the run during which it began to persist. After the checkpoint following [step 6.9.6d](#) of the run, no more instances of the ability can be created. Even if another run initiates during the reaction window following that checkpoint, new instances of any abilities persisting from the first run cannot become pending during the second run, even if their trigger conditions would normally be met during that second run.

Example: The Runner trashes AMAZE Amusements, causing its “Whenever a run on this server ends” ability to persist. When the run ends, the Runner uses Doppelgänger to make another run while the persisting ability is still pending. The ability remains active so that its existing pending instance can resolve, but another instance cannot be created even if the Runner steals an agenda during the second run.

10. Additional Rules

10.1. General

- 10.1.1. Some cards are UNIQUE, and have a unique symbol (◆) before their name to designate this. There can be only one unique card of the same name active at a time. If a unique card becomes active, any other card that shares its name is trashed during the next checkpoint. Trashing a card this way cannot be prevented. See [section 10.3](#).
- 10.1.2. To PURGE VIRUS COUNTERS is to remove all virus counters hosted on cards and return them to the bank. The Corp can purge virus counters as a basic action or with a card ability.
 - a. The Corp can always use a purge effect, even if there are no virus counters currently hosted on any cards. This is an exception to [rule 1.2.5](#).
- 10.1.3. Some abilities add a card to a player’s score area “as an agenda”. When this happens, the card loses all its previous properties and gains only those properties specified in the effect converting it. This conversion lasts until the card moves to a zone that is not a score area, at which point it returns to being its original printed card. If this happens in any way other than by agenda forfeit, the card is immediately trashed. See [rule 8.2.5](#).
- 10.1.4. Some abilities can convert a card into a counter. When this happens, the card loses all its previous properties and gains only those properties specified in the effect converting it. This conversion lasts until the counter moves to another zone, at which point it reverts to being a card, regains its original printed characteristics, and is trashed.

10.2. Information

The game, at its core, is about information. Much of the game revolves around deducing information about the opponent's cards and strategy.

10.2.1. Information in the game is classified as HIDDEN INFORMATION or OPEN INFORMATION.

10.2.2. Hidden Information

- a. HIDDEN INFORMATION is any information about the game, game state, or cards unavailable to one or more players. This includes facedown cards in play or in Archives, cards in HQ or R&D, cards in the Runner's grip or stack, etc.
- b. A player cannot learn hidden information without the aid of a game effect, rule, or another player verbally communicating the information. However, if a player that has access to information about the game or a card chooses to verbally share it with their opponent, that player is not required to tell the truth. Bluffing is allowed.

Example: The Runner uses Indexing. While rearranging cards, the Runner places Braintrust on top of R&D, followed by Snare!. The Corp draws at the beginning of their next turn, then spends their first click to draw a card. For their second click, the Corp installs a card in an empty remote server, telling the Runner that they should run it because it is a Braintrust.

10.2.3. Open Information

- a. OPEN INFORMATION is any information about the game, game state, cards, or abilities that is available to both players. This includes faceup cards in Archives and the heap, the number of cards in HQ, R&D, the stack, and the grip, the number of credits in a credit pool, and any other information continuously available to both players.
- b. Open information cannot be hidden from an opponent. A player must allow their opponent to discover the information themselves if they attempt to do so.

Example: The Runner installs Femme Fatale, choosing a piece of ice protecting HQ with Femme's ability. The Runner must explicitly state to the Corp which ice has been chosen and must continue to do so if the Corp asks for Femme Fatale's target during a later turn.

10.3. Checkpoints

10.3.1. A CHECKPOINT is a **process** wherein **objects** that have entered an illegal state are corrected, expired effects are removed, and other important conditions are checked. This procedure is automatically performed at several timing points during the game, and entails the following steps, which are performed in order:

- a. Each active conditional ability looks at the changes to the game state since the beginning of the last checkpoint to see if its condition has been met. Any ability that has met its condition creates the appropriate instances of itself and marks them as pending, as described in [section 9.6](#).
- b. Any ability with a duration that has passed is removed from the game state.

*Example: During the checkpoint that follows the end of an encounter, abilities that increased the strength of an **icebreaker** during that encounter expire and the **icebreaker's** strength is correspondingly lowered.*

Example: The Corp has Targeted Marketing in the play area when the Runner steals an agenda. In the next checkpoint, the game recognizes that the ability on Targeted Marketing that was preventing the operation from being trashed no longer applies, and so the Corp trashes Targeted Marketing as if it were completing its resolution.

- c. If the agendas in either player's score area total 7 or more agenda points, that player wins the game. If both players would win this way simultaneously, the game ends in a draw.
- d. If 2 or more unique (◆) cards with the same name are active, for each such name, all of those cards except the one that became active most recently are trashed. **If 2 or more console cards are installed under the control of the same player, for each such player, all of those cards except the one that was installed most recently are trashed.**
- e. If any **objects** break any restrictions of card abilities or the game rules (such as the Runner's memory limit) or are installed or hosted in an illegal location, an **appropriate** set of those **objects** are trashed. **A set is appropriate for this purpose if trashing the objects in the set would leave all remaining installed or hosted objects in legal locations and if no object can be removed from the set while maintaining that property.** If there are multiple distinct **appropriate** sets, and one player controls all the **objects** in **each of** those sets, that player chooses which set to trash. If the sets contain **objects** controlled by both players, the active player chooses **which set is trashed**.

Example: The Runner controls a Chisel hosted on an unrezzed piece of ice. The Corp rezzes that ice, revealing it to be Tithonium. Since Tithonium has an ability prohibiting cards from being hosted on it, Chisel is now installed in an illegal location and must be trashed.

*Example: The Runner has installed programs with total memory cost equal to their memory limit when the Corp plays Bad Times, reducing the memory limit by 2[MU]. The Runner can trash any two programs with a memory cost of 1[MU] **each**, or any one program with a memory cost of 2[MU] or greater. They cannot trash any 0[MU] programs, nor can they trash any set of more than one program with a greater total memory cost than 2[MU], since those **objects can be removed from the trashed set while maintaining that the result will be a legal game state.***

- f. Any **objects** that were hosted on an agenda that moved from a score area to any zone other than a score area are trashed.
 - g. Any **objects** that were hosted on an installed card that was uninstalled are trashed, except for cards or counters that are temporarily in the set aside zone because of [rule 9.5.5](#). This step is repeated until no more cards or counters are trashed.
 - h. Any remote server with no cards protecting it, **in its root**, or in the process of being installed with a destination protecting it **or in its root** ceases to exist.
 - i. Any cards in discard piles that had been converted into counters or agendas return to their printed characteristics.
 - j. Any counters in a discard pile are returned to the bank.
- 10.3.2. After a checkpoint is completed, if it marked any **instances of** conditional abilities as pending, a new reaction window immediately opens to handle those abilities, even if this takes place during another reaction window. After a reaction window closes, the game proceeds to whatever was due to occur after the checkpoint that caused it to open.
- 10.3.3. Whenever a player would receive priority, first a checkpoint occurs. See also [section 9.2.4](#).
- 10.3.4. Whenever a player pays a cost, a checkpoint occurs immediately afterward, before continuing with the effect or timing structure in which the cost was paid. See also [section 1.16](#).
- 10.3.5. Whenever an instruction finishes resolving, a checkpoint occurs immediately afterward, before the next instruction becomes imminent. See also [section 9.11](#).**
- 10.3.6. The checkpoint following the last step of a timing structure takes place outside of that timing structure, as does any associated reaction window.**

Example: The Runner, playing as Jesminder Sareen, steals an agenda during a run on a server with AMAZE Amusements rezzed in its root. Since AMAZE Amusements's ability meets its trigger condition when the run ends, it will become pending in the reaction window following [step 6.9.6d](#) and attempt to give the Runner 2 tags. During this window, the run timing structure has closed, and Jesminder's interrupt ability cannot apply to those tags.

10.4. Damage

- 10.4.1. Many cards and ice subroutines inflict damage on the Runner. The Runner SUFFERS (sometimes referred to as "TAKES") damage according to the rules for the type of damage specified. If a Corp card "does" damage to the Runner, they suffer that much damage, with the additional provision that the Corp and the source card are responsible for that damage. If a card directs or allows the Runner to "suffer" damage, then the Runner and the source card are responsible for that damage.

Example: The Corp, playing as Argus Security, has scored The Cleaners. The Runner makes a run on R&D and steals Hostile Takeover. Argus Security's ability triggers and the Runner chooses to suffer 2 meat damage. As The Cleaners only adds additional damage to damage done by the Corp, its ability does not apply. When the Corp plays Punitive Counterstrike on their next turn, The Cleaners does increase the amount of damage done by Punitive Counterstrike.

- 10.4.2. The three types of damage the Runner can suffer are meat, net, and brain damage.
- a. MEAT DAMAGE and NET DAMAGE are resolved the same way. For each point of damage suffered, one card is randomly trashed from the grip.
 - b. For each point of BRAIN DAMAGE suffered, one card is randomly trashed from the grip and the Runner's maximum hand size is permanently reduced by 1. The Runner takes a brain damage token for each brain damage suffered to track the reduction in hand size.
- 10.4.3. If the Runner suffers more than 1 damage of any type, the cards are randomly trashed simultaneously.

Example: The Runner has 2 tags and four cards in their grip, one of which is I've Had Worse. The Corp plays BOOM!, dealing 7 meat damage to the Runner. If the Runner cannot or does not prevent at least 3 of the incoming damage, they immediately flatline as they have suffered more damage than they have cards in grip, and I've Had Worse does not have the opportunity to resolve. If the Runner does prevent enough of the damage to survive, the Runner randomly trashes as many cards as the remaining amount of damage. After all the cards have been randomly selected and placed in the heap, I've Had Worse meets its trigger condition if it is among those cards trashed by the damage.

- a. If an effect modifies the procedure for dealing damage such that the cards trashed cannot be selected simultaneously, then the cards are selected sequentially but are still trashed simultaneously.

Example: The Corp is playing as Chronos Protocol: Selective Mind Mapping when the Runner accesses a Snare! before taking any other damage during this turn. The Corp looks at the grip and selects a card to be trashed as the first point of net damage for the turn, then two other cards are randomly selected. All 3 cards are trashed simultaneously.

- 10.4.4. If the Runner takes more damage than the number of cards in their grip, or if they have a maximum hand size of less than zero at the end of their turn, then they are FLATLINED and the Corp wins the game.

10.5. Tags

Tags represent the Corp's possession of valuable information about the Runner, such as a data trail they've left behind or the physical location they are jacked in from.

- 10.5.1. A TAG refers to a tag token placed on the Runner. Tags are used in card abilities and basic actions.

- 10.5.2. The Runner is TAGGED if there are one or more tags on them.
- 10.5.3. While the Runner is tagged, the Corp can, as an action, spend [click] and 2[c] to trash one of the Runner's resource cards.
- 10.5.4. While the Runner is tagged, they can, as an action, spend [click] and 2[c] to remove one tag, returning it to the bank.
- 10.5.5. The Runner controls each tag on them, but the Corp can pay costs involving tags as if they controlled the Runner's tags.

Example: The Runner has a tag, and makes a run on a server with Keegan Lane installed. Even though the tag token is controlled by the Runner, the Corp can remove it to pay the cost of Keegan Lane's ability to trash one of the Runner's programs.

10.6. Bad Publicity

Nobody likes corrupt Corps, or at least, most people say they don't. If a Corp has a bad reputation, the Runner has an easier time finding help to attack that Corp's servers. This extra help is abstracted as extra credits the Runner gets to use during runs.

- 10.6.1. BAD PUBLICITY refers to a bad publicity token placed on the Corp. Bad publicity is used by card abilities and during runs.
- 10.6.2. For each bad publicity the Corp has, the Runner gains 1[c] at the beginning of each run, during [step 6.9.1c](#). Any unspent bad publicity credits return to the bank at the end of the run, during [step 6.9.6a](#).
 - a. If the Corp takes bad publicity while a run is already underway, the Runner does not gain additional bad publicity credits for that run.
- 10.6.3. The Corp controls each bad publicity on them, but the Runner can pay costs involving bad publicity as if they controlled the Corp's bad publicity.

10.7. Link

The Runner's link is the number of points their path through the Net traverses between their rig and the Corp's server. The more proxies, redirects, and other intermediary connecting points between the Corp and the Runner, the harder it is for sysops to track down the Runner's virtual location.

- 10.7.1. The Runner's LINK VALUE is calculated by adding the base link on their identity card to the link ([link]) produced by their installed cards.
- 10.7.2. Link is primarily used to contest traces, as described in [section 10.8](#), but other abilities can also refer to the Runner's link value.

10.8. Traces

A trace is the Corp's attempt to connect from their servers back to the Runner. Many traces can allow the Corp to gain information about the Runner, tagging them, but traces can also produce a wide variety of other effects. Traces most commonly originate from ice subroutines, but Corporate operations can produce some of the nastiest trace effects.

- 10.8.1. Some card abilities initiate a TRACE ATTEMPT (sometimes simply called a TRACE) on the Runner. Traces are marked by the language "Trace [N]" on a card, where the given value N is the BASE TRACE STRENGTH of the trace. A trace pits the Corp's trace strength against the Runner's link strength, both of which can be increased by spending credits.
- 10.8.2. The Corp acts first during a trace attempt, openly spending any number of credits. This sets the TRACE STRENGTH, which is the base trace strength plus the number of credits spent.
- 10.8.3. After the Corp spends their credits, the Runner has the opportunity to spend credits to establish their link strength. The Runner's LINK STRENGTH is equal to their link value plus the number of credits spent.
- 10.8.4. After the Runner's link strength is established, it is compared to the Corp's trace strength. If the trace strength exceeds the link strength, the trace attempt is successful and any "If successful" abilities associated with the trace meet their trigger conditions. If the link strength is equal to or greater than the trace strength, then the trace attempt is unsuccessful, and any "If unsuccessful" abilities associated with the trace meet their trigger conditions.
- 10.8.5. "If successful" and "if unsuccessful" effects following the "Trace [N]" indicator are conditional abilities tied to the results of the trace. Any instructions in the body of the trace without an "if successful" or "if unsuccessful" are conditional abilities with the implicit trigger condition, "when the trace is determined to be successful or unsuccessful."

Example: Gemini's subroutine initiates a trace attempt with two associated conditional abilities. The first is "If successful, do 1 net damage." and the second is "When the trace is determined to be successful or unsuccessful, if your trace strength is 5 or greater, do 1 net damage." Both of these will become pending (if appropriate) [after step 10.8.6e](#) below.

10.8.6. Steps of Resolving a Trace Attempt

- a. The trace initiates. At this time, any "when a trace is initiated" abilities meet their trigger conditions.
- b. A checkpoint occurs.
- c. The Corp may spend credits to increase the trace strength.
- d. The Runner may spend credits to increase their link strength.

- e. The trace is determined to be successful or unsuccessful.
- f. The trace is complete.

10.9. Load and Empty

- 10.9.1. To LOAD a card with counters is to place those counters on that card. A card is EMPTY when it is no longer hosting any counters of a type that were previously loaded onto it.
- 10.9.2. An ability on a card that refers to that card becoming empty is always linked to a preceding ability on that same card that loaded it with counters.

Example: The Runner installs Crowdfunding, and its first ability becomes pending. Even though there are no credits hosted on Crowdfunding until that ability resolves, the following "When it is empty, ..." ability does not meet its trigger condition because the card has not yet been loaded.

- 10.9.3. If any other types of counters are placed on that card without having been loaded, and those counters are subsequently removed, then empty abilities on that card do not meet their condition.
- 10.9.4. Loading a card with counters does not impose any further restrictions upon those counters for that card. Other abilities can still issue instructions to act upon counters of that type, including but not limited to placing more of them or removing some of them.

10.10. Infinite Loops

- 10.10.1. If a mandatory infinite loop is created (a player cannot choose to stop resolving the loop) then the player who is resolving the loop chooses a number. The loop instantaneously resolves that many times, and then ends.

Example: The Runner runs into a rezzed Wormhole. The only other piece of ice that is rezzed is a Wormhole, and so a mandatory infinite loop is created where each of the Wormholes' subroutines resolves the other. The Corp chooses how many times this loop occurs, say 2,157 times, and then the Runner continues the run.

- 10.10.2. If an optional infinite loop is created (a player can choose to stop resolving the loop) during a run, then the Runner must jack out unless another card ability prevents them from doing so. If the Runner cannot jack out, then it is the Corp's responsibility to end the loop by letting the Runner continue the run.

10.11. Self-Referential Language

- 10.11.1. If the card name referenced on a card matches the name of that card, and the reference is not plural and/or does not include an additional modifier (e.g., "other copies" or "all"), then the card name only refers to that copy of the card, and not to any other copies of it.
- 10.11.2. If a card copies the text of another card, and the copied text includes a self-reference, the copied text does not function unless it explicitly says so.

Example: The Runner uses Media Blitz to copy the text of a Private Security Force in the Runner's score area. Because the self-reference in Private Security Force does not match the name of the card Media Blitz, the ability has no effect.

11. Appendix: Timing Structure Reference

11.1. Overview

- 11.1.1. The following pages contain quick reference guides for the main timing structures in the game. When initiating a new timing structure, begin at the first numbered step and proceed through the structure sequentially, unless instructed to move to a different phase or step.
- 11.1.2. These guides are intended to serve as shorthand references only; for comprehensive rules relating to each timing structure, see the accompanying rules sections.

11.2. Timing Structure of Turns

11.2.1. Corp's Turn ([Section 5.6](#))

1) Draw Phase

- a) The Corp gains allotted clicks.
- b) Paid ability window: (P) (R) (S).
- c) The Corp's recurring credits refill.
- d) The Corp's turn begins.
- e) The Corp draws 1 card.

2) Action Phase

- a) Paid ability window: (P) (R) (S).
- b) Does the Corp have unspent clicks?
 - i) If yes, the Corp takes an action.
 - ii) If no, go to (3).
- c) Return to (a).

3) Discard Phase

- a) The Corp discards cards.
- b) Paid ability window: (P) (R).
- c) The Corp loses unspent [click].
- d) The Corp's turn ends.
- e) The Corp's turn is complete, and the game moves to the Runner's turn.

11.2.2. Runner's Turn ([Section 5.7](#))

1) Action Phase

- a) The Runner gains allotted clicks.

- b) Paid ability window: (P) (R).
- c) The Runner's recurring credits refill.
- d) The Runner's turn begins.
- e) Paid ability window: (P) (R).
- f) Does the Runner have unspent clicks?
 - i) If yes, the Runner takes an action.
 - ii) If no, go to (2).
- g) Return to (e).

2) Discard Phase

- a) The Runner discards cards.
- b) Paid ability window: (P) (R).
- c) The Runner loses unspent [click].
- d) The Runner's turn ends.
- e) The Runner's turn is complete, and the game moves to the Corp's turn.

11.3. Timing Structure of a Run ([Section 6.9](#))

1) Initiation Phase

- a) The Runner announces **the** attacked server.
- b) The Runner gains bad publicity credits.
- c) The run begins.
- d) Is there ice protecting server?
 - i) If yes, go to (2).
 - ii) If no, go to (4).

2) Approach Ice Phase

- a) The Runner approaches ice.
- b) Paid ability window: (P) (R) and ice can be rezzed.
- c) Is the approached ice rezzed?
 - i) If yes, go to (3).
 - ii) If no, go to (4).

3) Encounter Ice Phase

- a) The Runner encounters ice.
- b) Paid ability window: (P) and subroutines can be broken.
- c) **Are there unbroken subroutines to resolve?**
 - i) **If yes, the Corp resolves the next one.**
 - ii) **If no, go to (e).**
- d) **Return to (c).**
- e) Go to (4).

4) Movement Phase

- a) **If the run got here from (2) or (3), the Runner passes ice.**
- b) **Paid ability window: (P).**
- c) **The Runner may jack out.**
- d) **The Runner moves 1 position inward, if possible.**

- e) **Paid ability window: (P) (R).**
 - f) **Did the Runner move to a new position?**
 - i) **If yes, go to (2).**
 - ii) **If no, go to (g).**
 - g) **The Runner approaches the server.**
 - h) **Go to (5).**
- 5) Success Phase**
- a) The run is declared successful.
 - b) The Runner **breaches the attacked server.**
 - c) Go to (6).
- 6) Run Ends Phase**
- a) Close or resolve priority windows from before “end the run”.
 - b) The Runner loses unspent bad publicity credits.
 - c) If applicable, the run is declared unsuccessful.
 - d) The run is complete.

11.4. Timing Structure of Breaching a Server ([Section 7.5](#))

- 1) **The breach** begins.
- 2) If **breaching** Archives, facedown cards in Archives are turned faceup.
- 3) **If breaching HQ or R&D, determine how many accesses from Corp’s hand or deck.**
- 4) Are there **candidate cards** remaining **to access**?
 - a) If yes, **the Runner chooses a candidate.**
 - b) If no, go to (7).
- 5) **The Runner accesses the chosen card.**
- 6) Return to (4).
- 7) Breaching the server is complete.

11.5. Timing Structure of Accessing a Card ([Section 7.2](#))

- 1) Card is accessed.
- 2) The Runner may trash card **or use another mid-access ability.**
- 3) If card is an agenda, the Runner steals it.
- 4) Access is complete.

Acknowledgements

- *Netrunner* Original Game Design: Richard Garfield
- *Android: Netrunner*
 - Game Development: Lukas Litzsinger
 - Expansion Development: Lukas Litzsinger, Damon Stone, and Michael Boggs
 - Rules by: Adam Baker, Michael Boggs, and Erik Dahlman
 - Android Universe created by: Kevin Wilson with Daniel Lovat Clark
- NISEI
 - Rules Manager: Jade Wesley
 - Assistant Rules Manager: Jamie Perconti
 - Rules Associates: **Ruben P. Pieters, Justin Prentice, and** Tim Vaduva
 - Rules Editor: Kayli Ammen
 - **Additional Contributions: Pat Chapman and Noah Bogart**

Netrunner is a ™ of R. Talsorian Games, Inc. Android is ™ & © Fantasy Flight Games.

Although these rules are made to be compatible with cards from *Android: Netrunner*, they are not in any way associated with or endorsed by Fantasy Flight Games, R. Talsorian Games, or Wizards of the Coast.