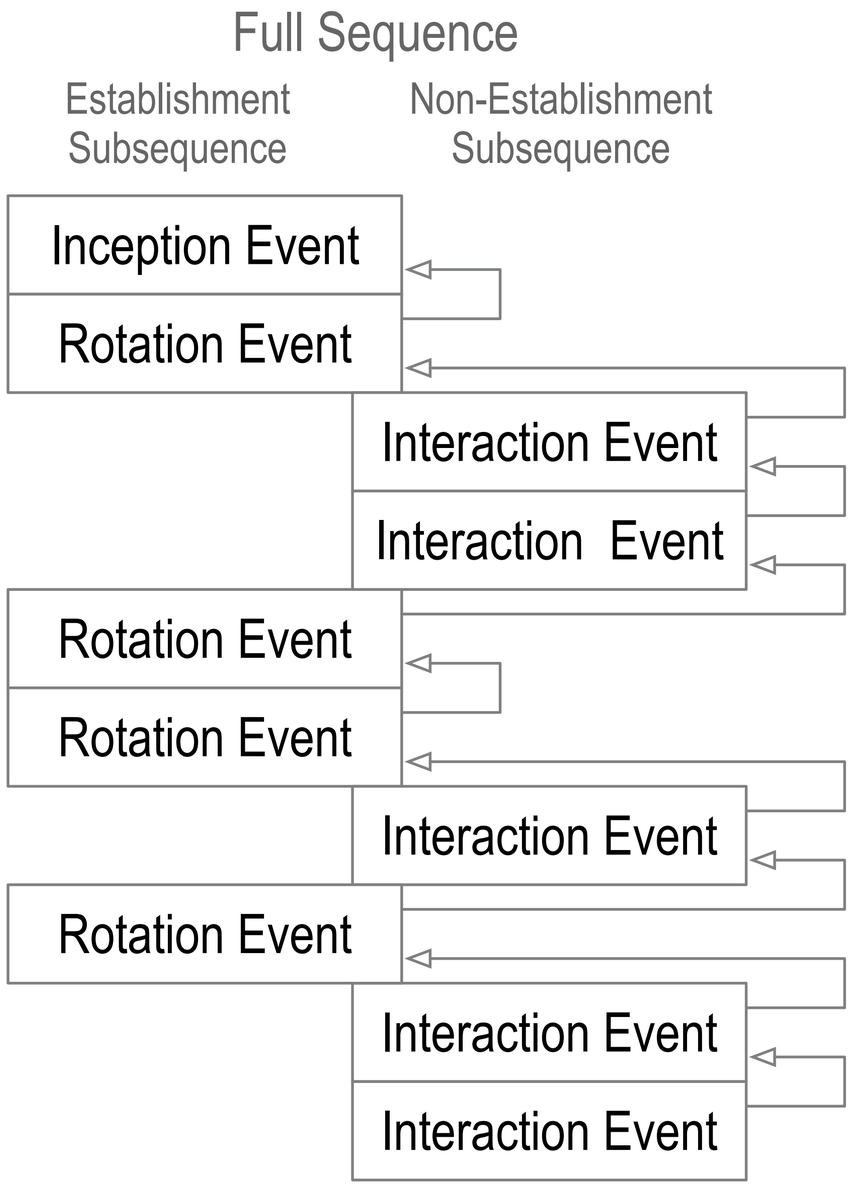
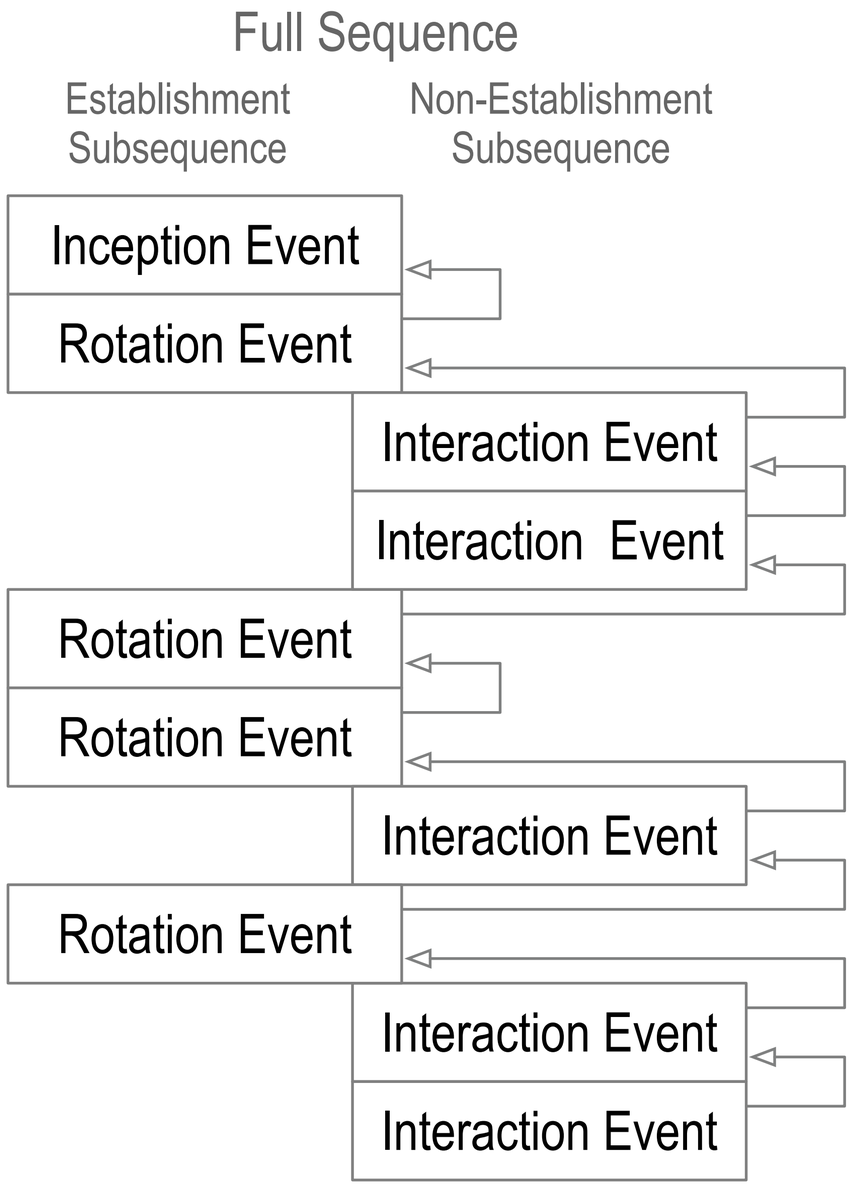
The Duplicity Game:
or why you can trust KERI

Samuel M. Smith Ph.D.
sam@prosapien.com
IIW Spring 2020 April 28-30
version 2.25
https://github.com/SmithSamuelM/Papers
https://github.com/SmithSamuelM/Papers/blob/master/presentations/KERI2_Overview_IIW_2020_A.pdf
https://github.com/SmithSamuelM/Papers/blob/master/whitepapers/KERI_WP_2.x.web.pdf
https://github.com/decentralized-identity/keri
https://github.com/decentralized-identity/keri/blob/master/implementation.md
https://hackmd.io/orhyiJkLT721v4PCPkvQiA?both

Inconsistency and Duplicity

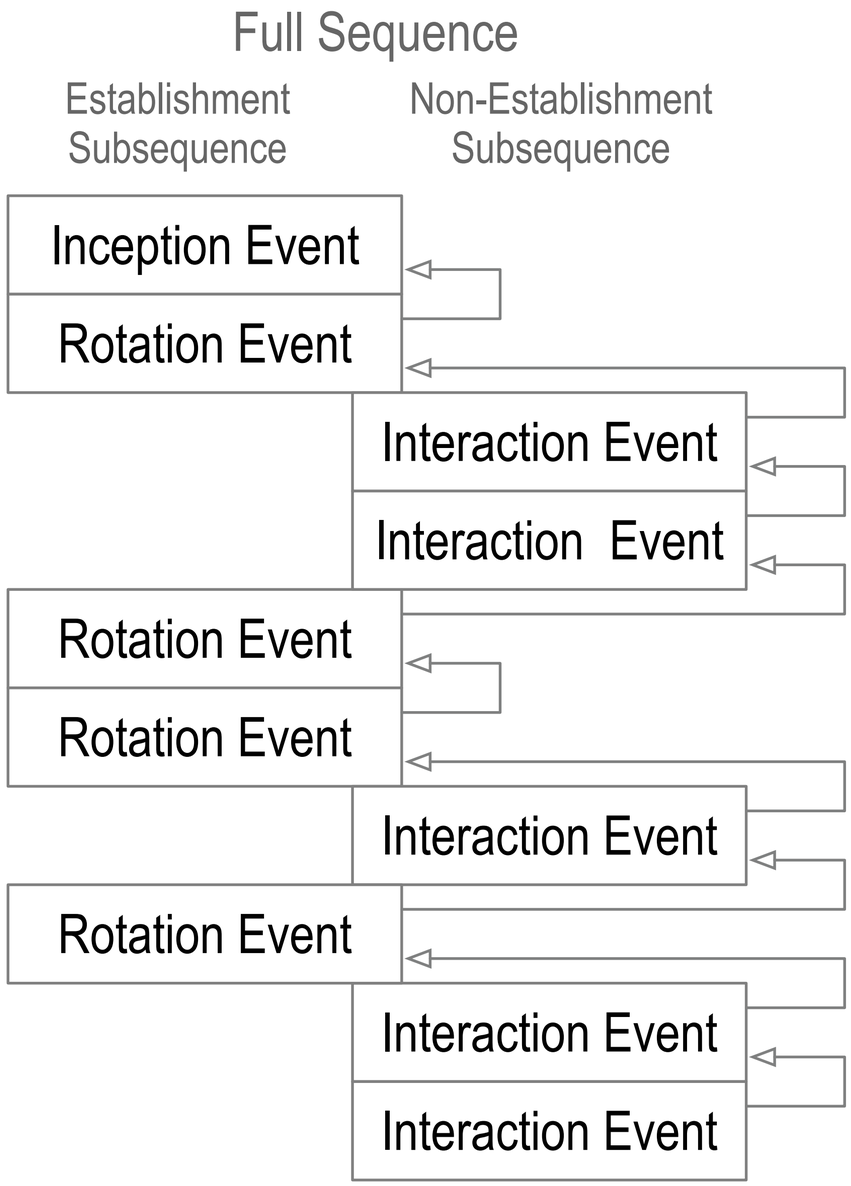


Inconsistency and Duplicity



Inconsistency vs. Duplicity

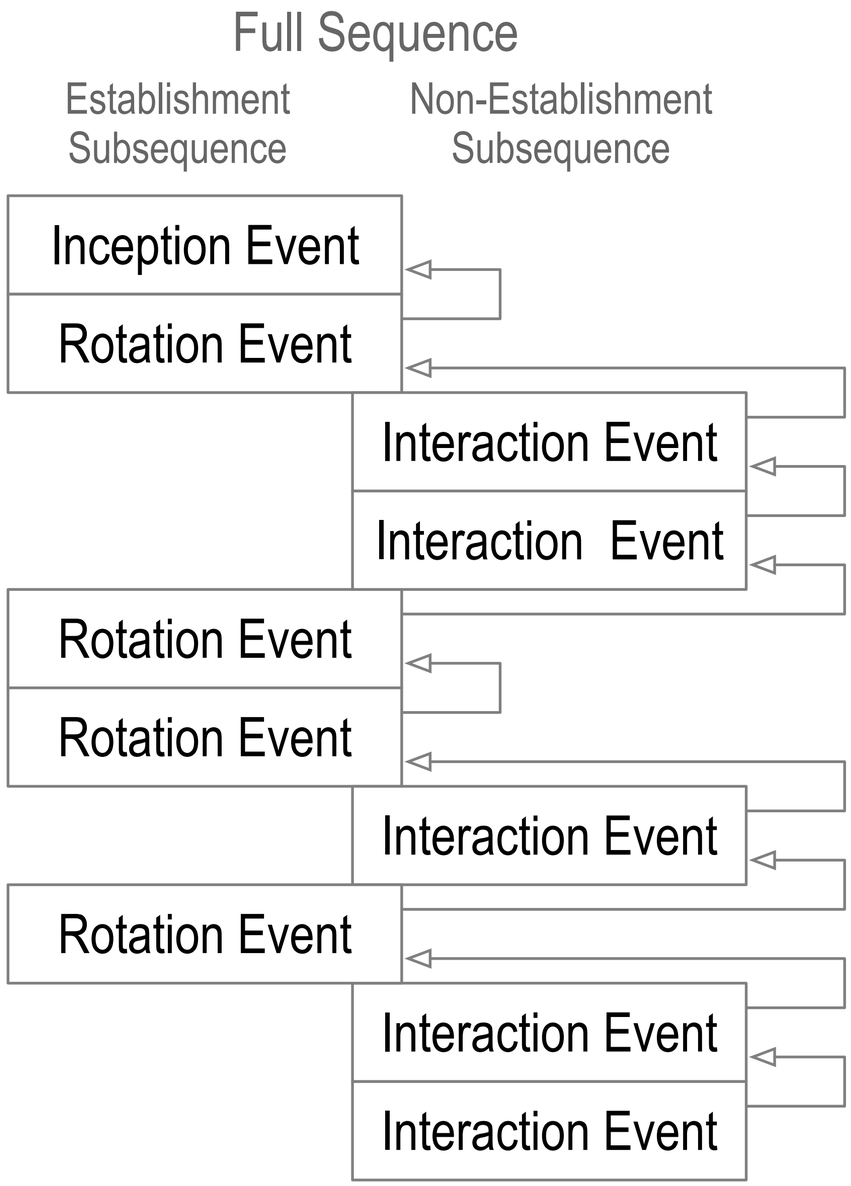
Inconsistency and Duplicity



Inconsistency vs. Duplicity

inconsistency: lacking agreement, as two or more things in relation to each other

Inconsistency and Duplicity

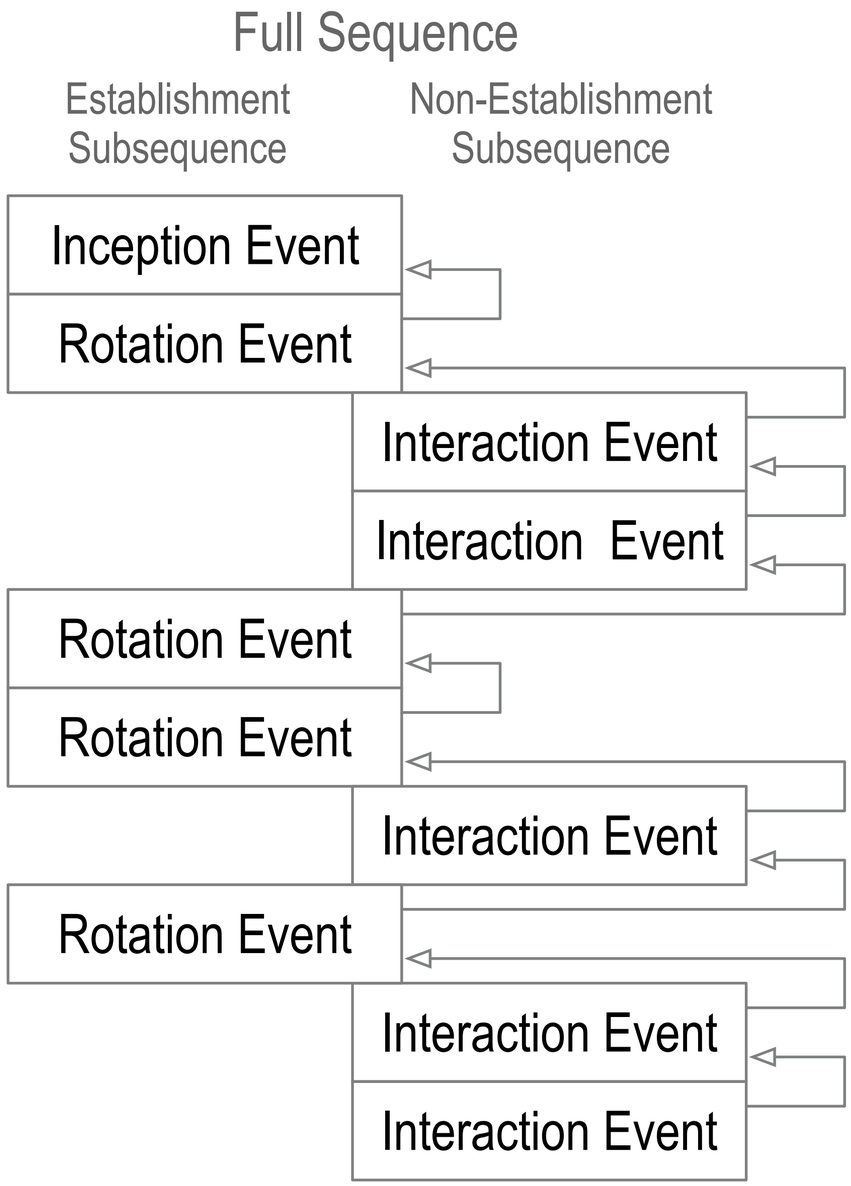


Inconsistency vs. Duplicity

inconsistency: lacking agreement, as two or more things in relation to each other

duplicity: acting in two different ways to different people concerning the same matter

Inconsistency and Duplicity



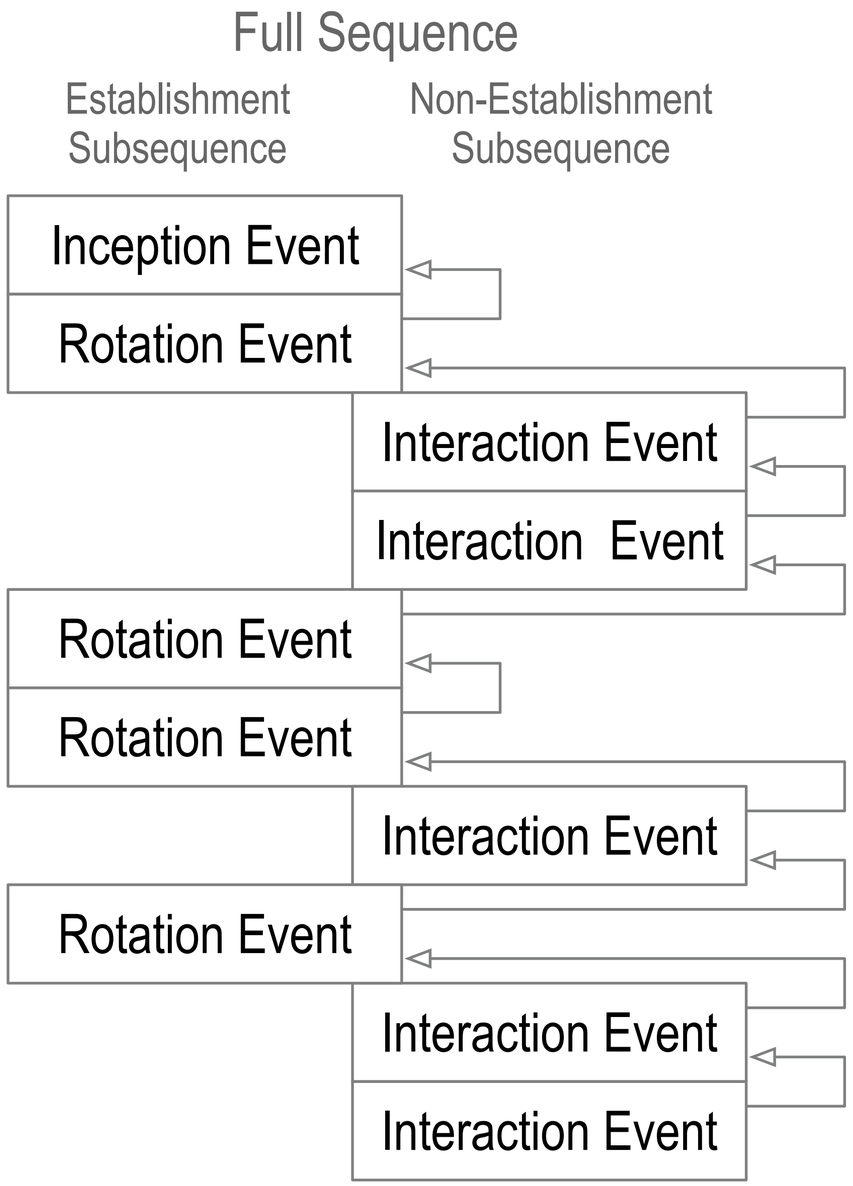
Inconsistency vs. Duplicity

inconsistency: lacking agreement, as two or more things in relation to each other

duplicity: acting in two different ways to different people concerning the same matter

Internal vs. External Inconsistency

Inconsistency and Duplicity



Inconsistency vs. Duplicity

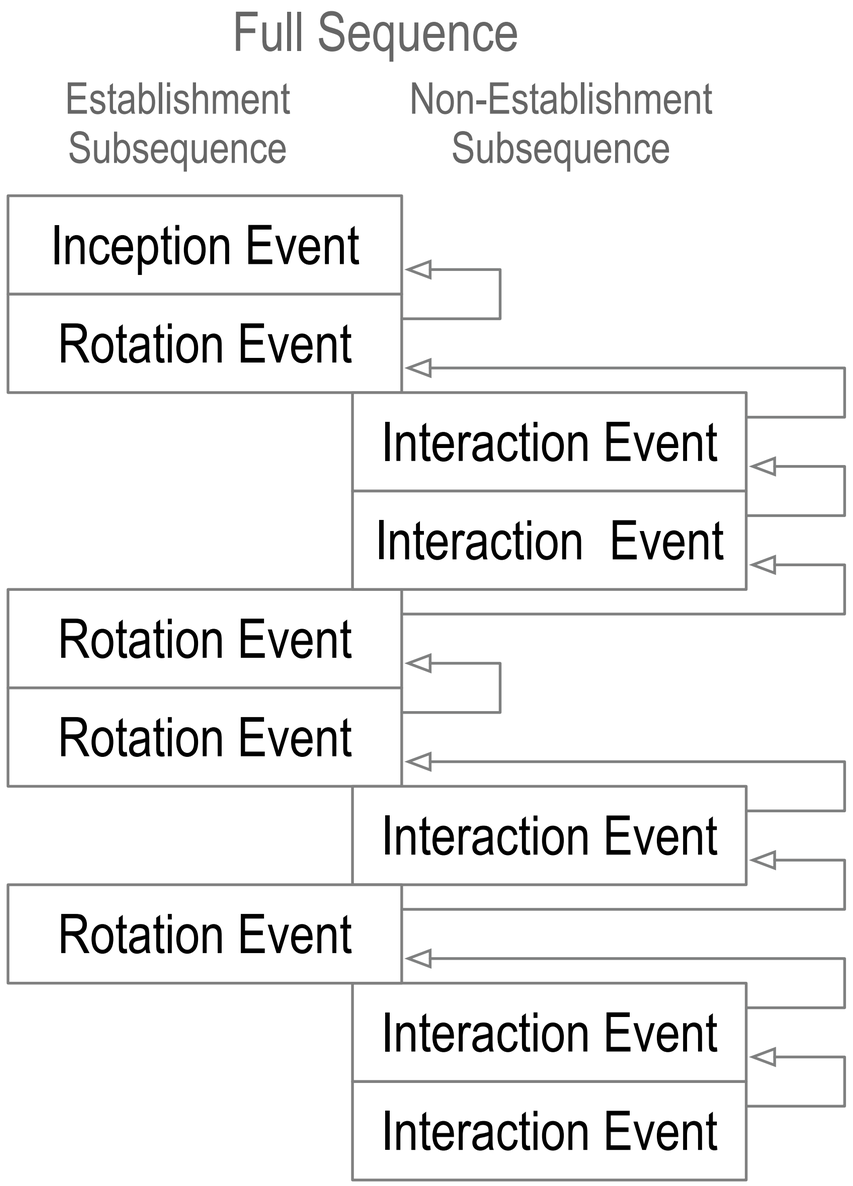
inconsistency: lacking agreement, as two or more things in relation to each other

duplicity: acting in two different ways to different people concerning the same matter

Internal vs. External Inconsistency

Internally inconsistent log = not verifiable.

Inconsistency and Duplicity



Inconsistency vs. Duplicity

inconsistency: lacking agreement, as two or more things in relation to each other

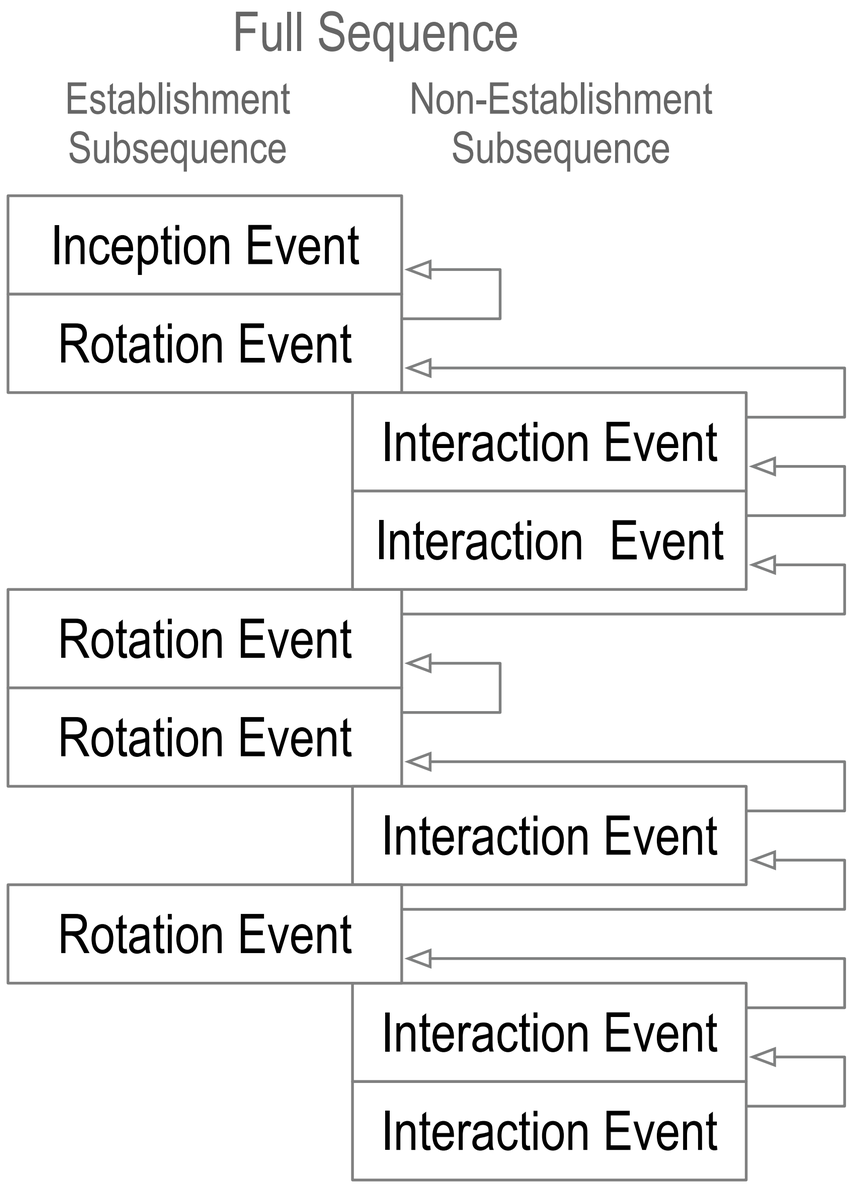
duplicity: acting in two different ways to different people concerning the same matter

Internal vs. External Inconsistency

Internally inconsistent log = not verifiable.

Log verification from self-certifying root-of-trust protects against internal inconsistency.

Inconsistency and Duplicity



Inconsistency vs. Duplicity

inconsistency: lacking agreement, as two or more things in relation to each other

duplicity: acting in two different ways to different people concerning the same matter

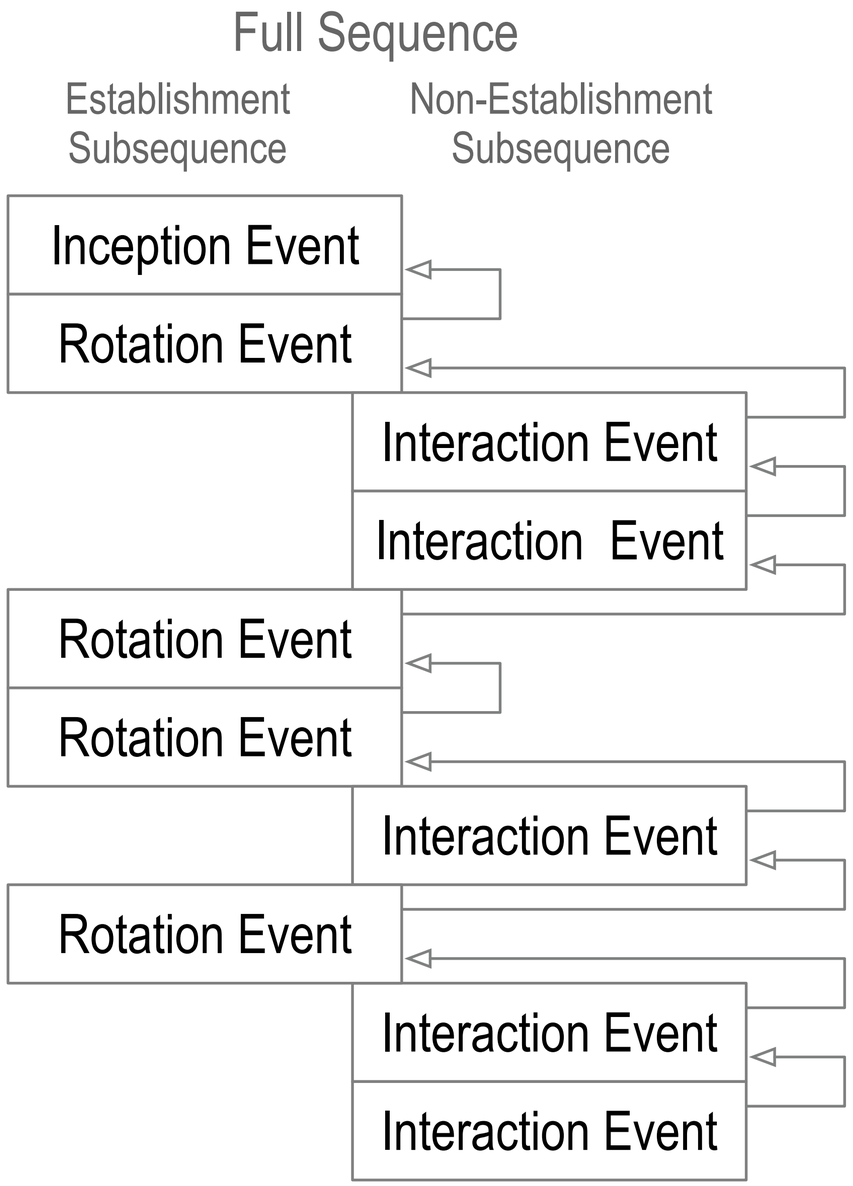
Internal vs. External Inconsistency

Internally inconsistent log = not verifiable.

Log verification from self-certifying root-of-trust protects against internal inconsistency.

Externally inconsistent log with a purported copy of log but both verifiable = duplicitous.

Inconsistency and Duplicity



Inconsistency vs. Duplicity

inconsistency: lacking agreement, as two or more things in relation to each other

duplicity: acting in two different ways to different people concerning the same matter

Internal vs. External Inconsistency

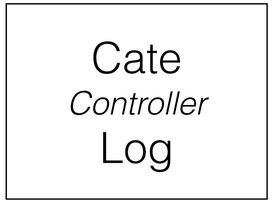
Internally inconsistent log = not verifiable.

Log verification from self-certifying root-of-trust protects against internal inconsistency.

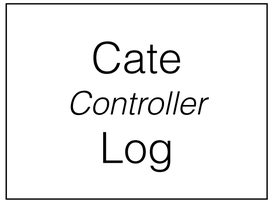
Externally inconsistent log with a purported copy of log but both verifiable = duplicitous.

Duplicity detection protects against external inconsistency.

Duplicity Game

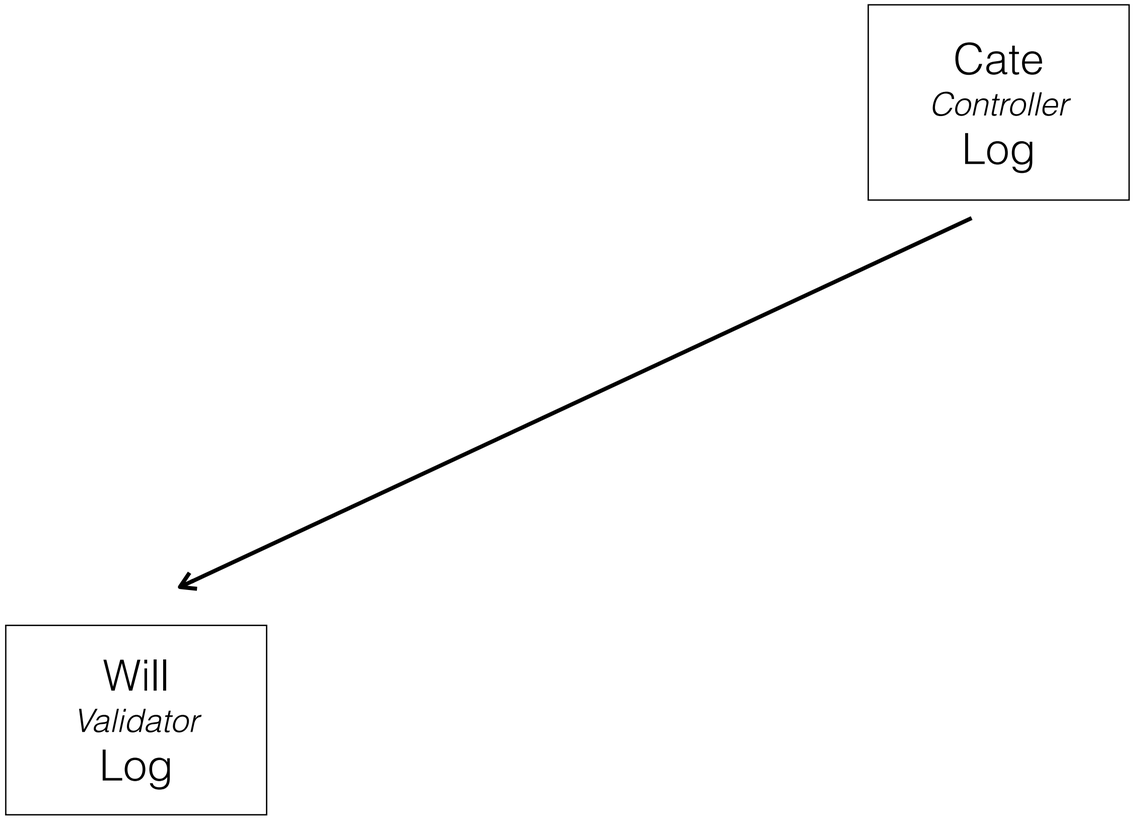


Duplicity Game



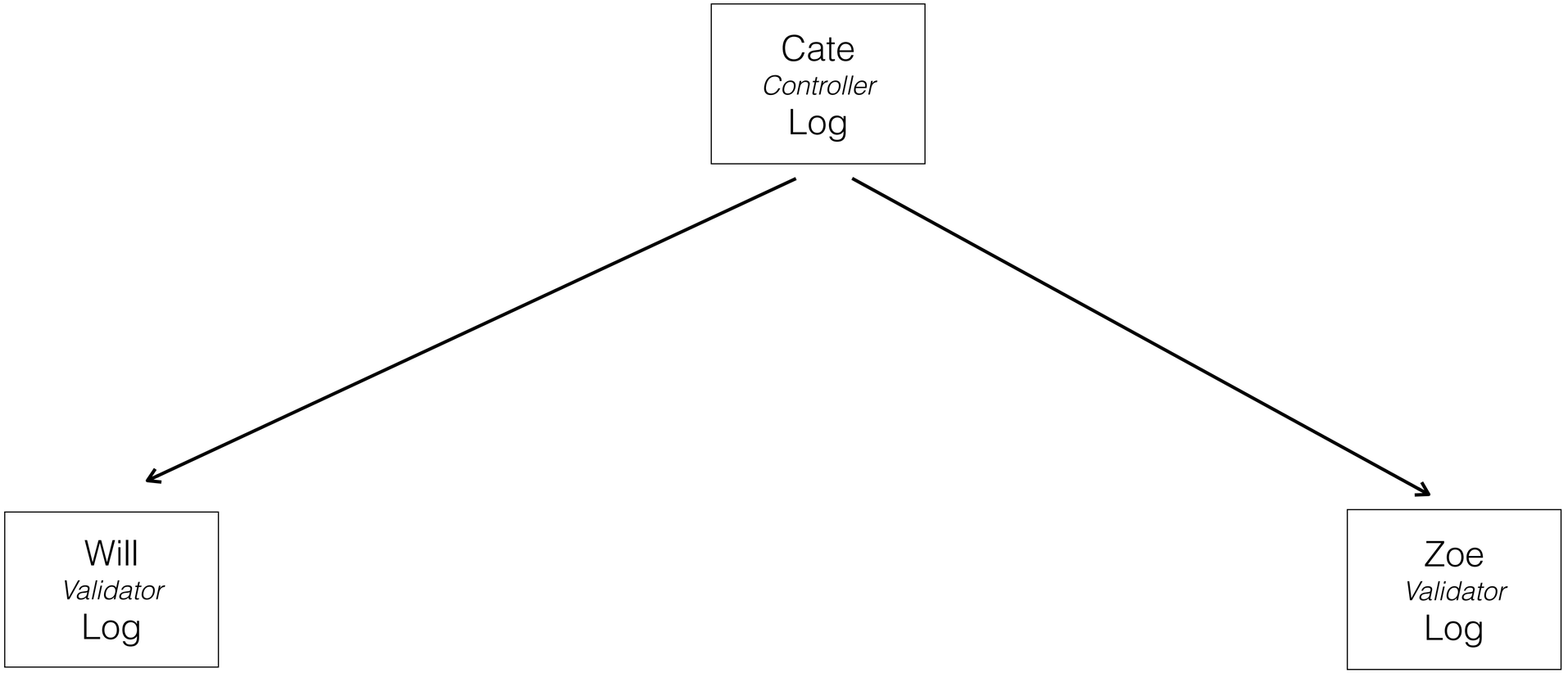
private (one-to-one) interactions

Duplicity Game



private (one-to-one) interactions

Duplicity Game

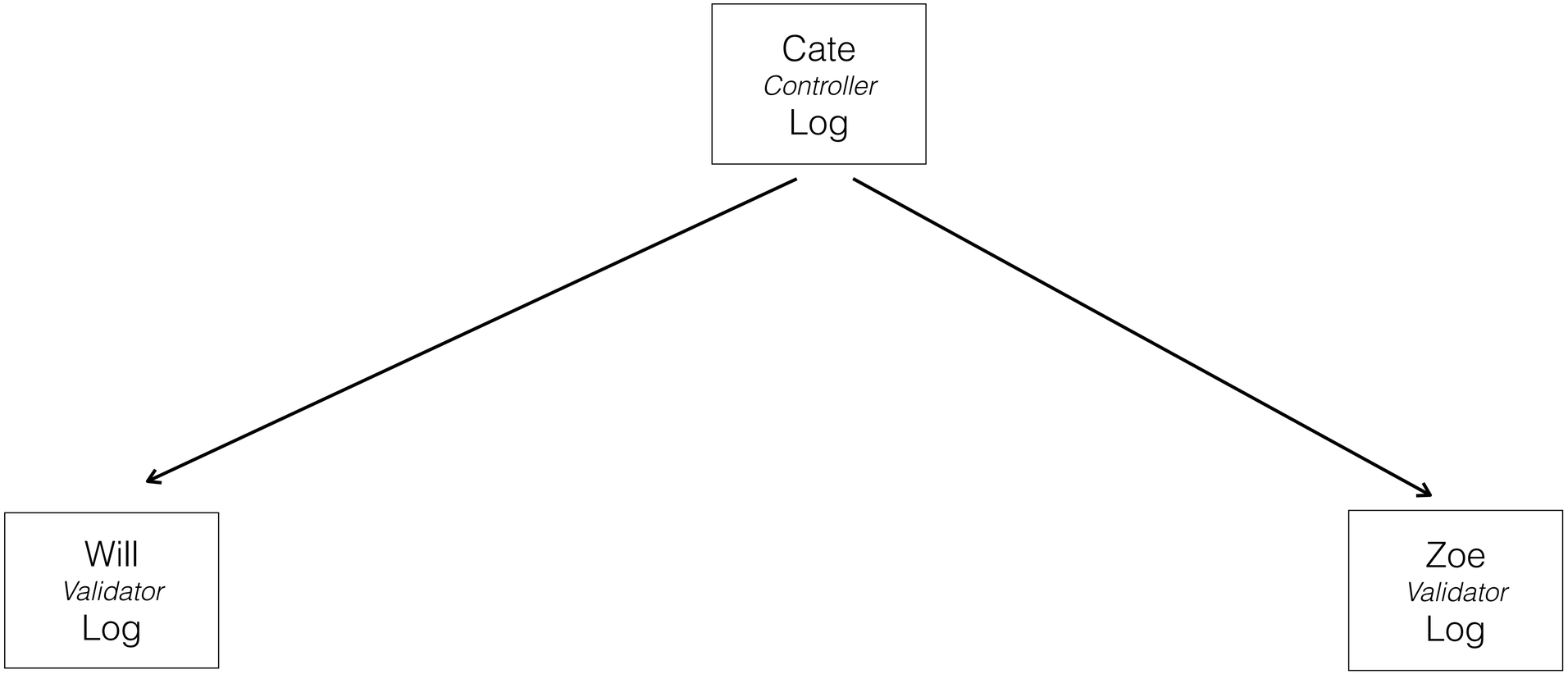


private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

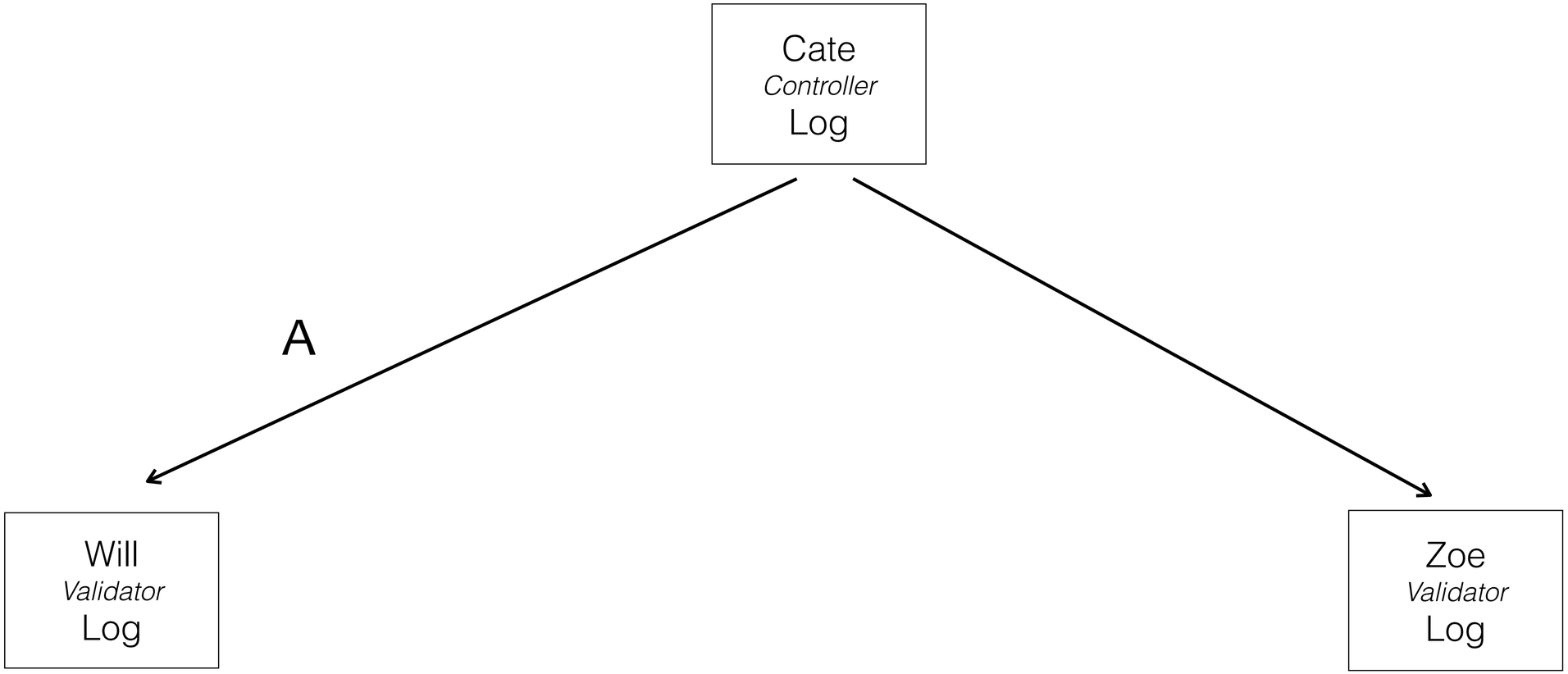


private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

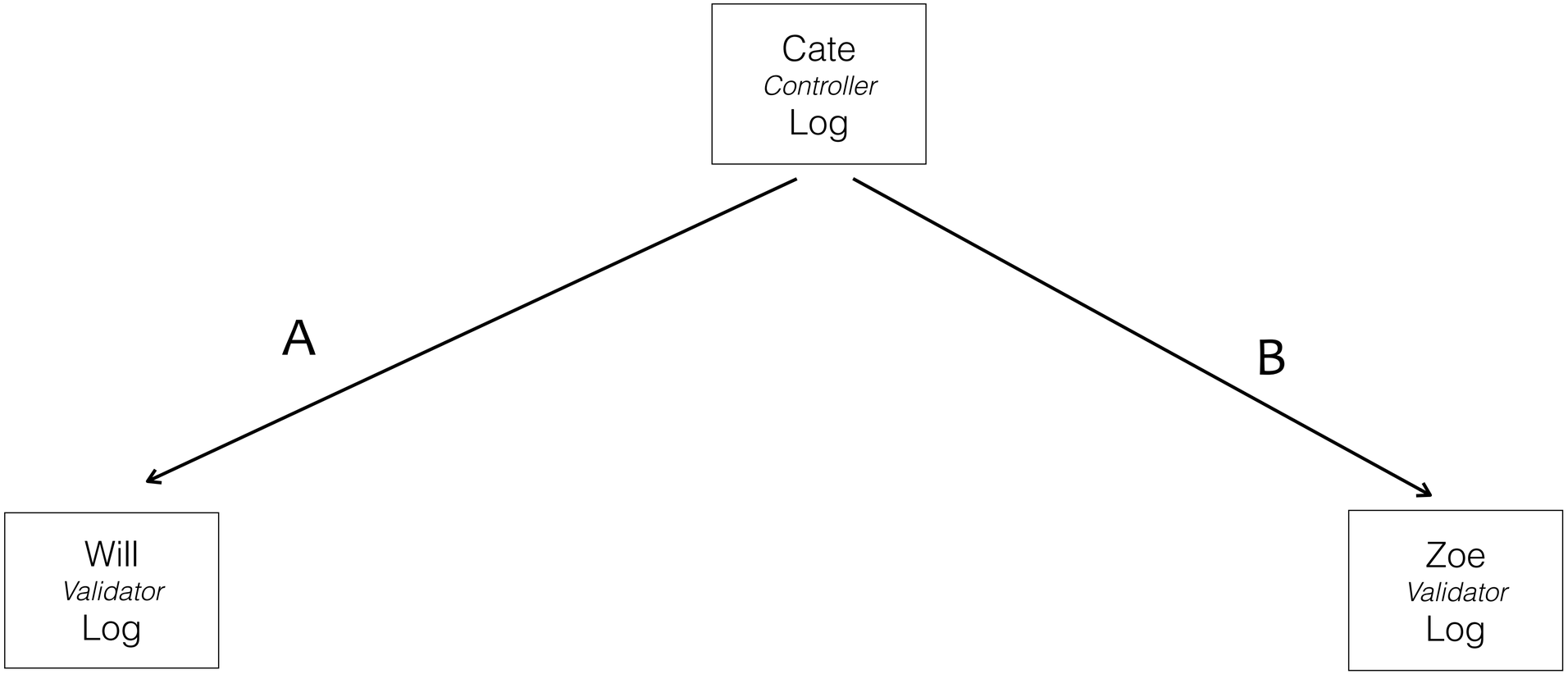


private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee



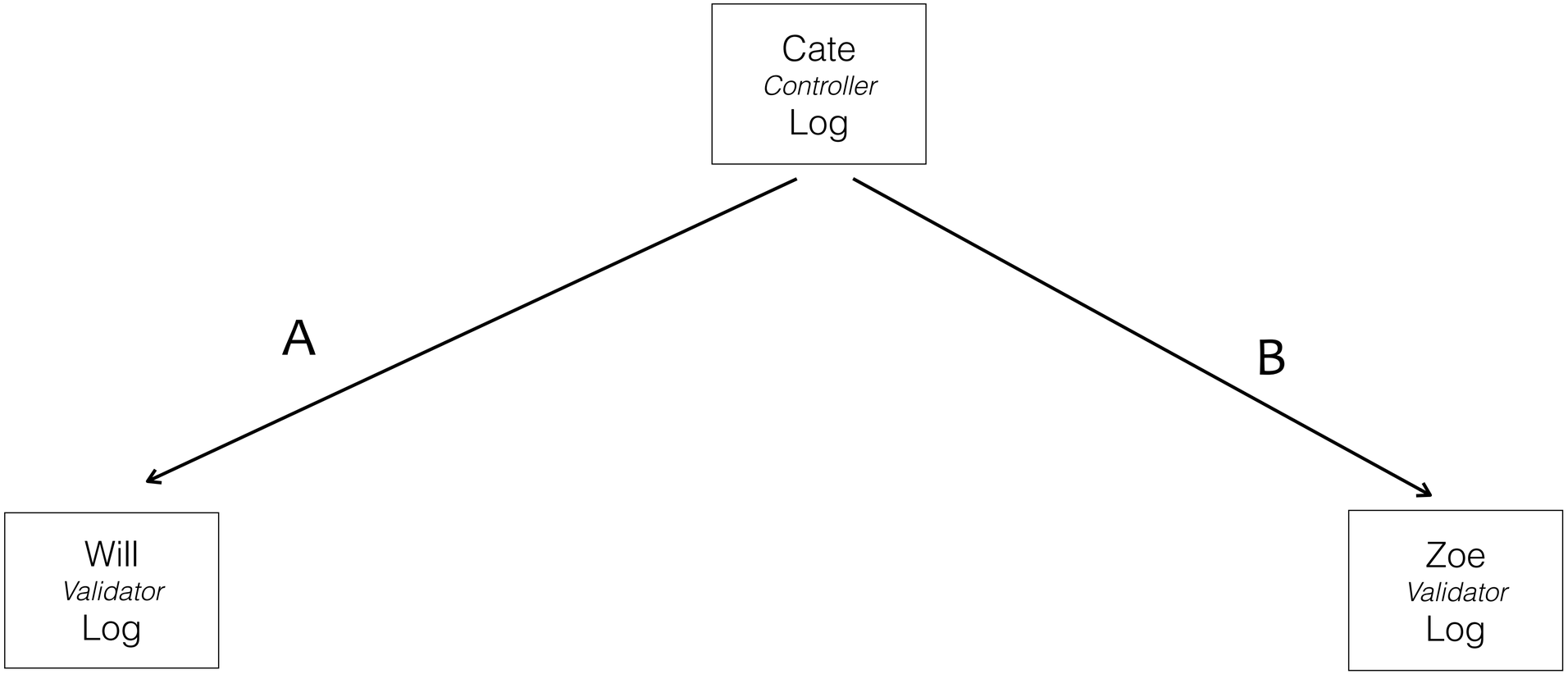
private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



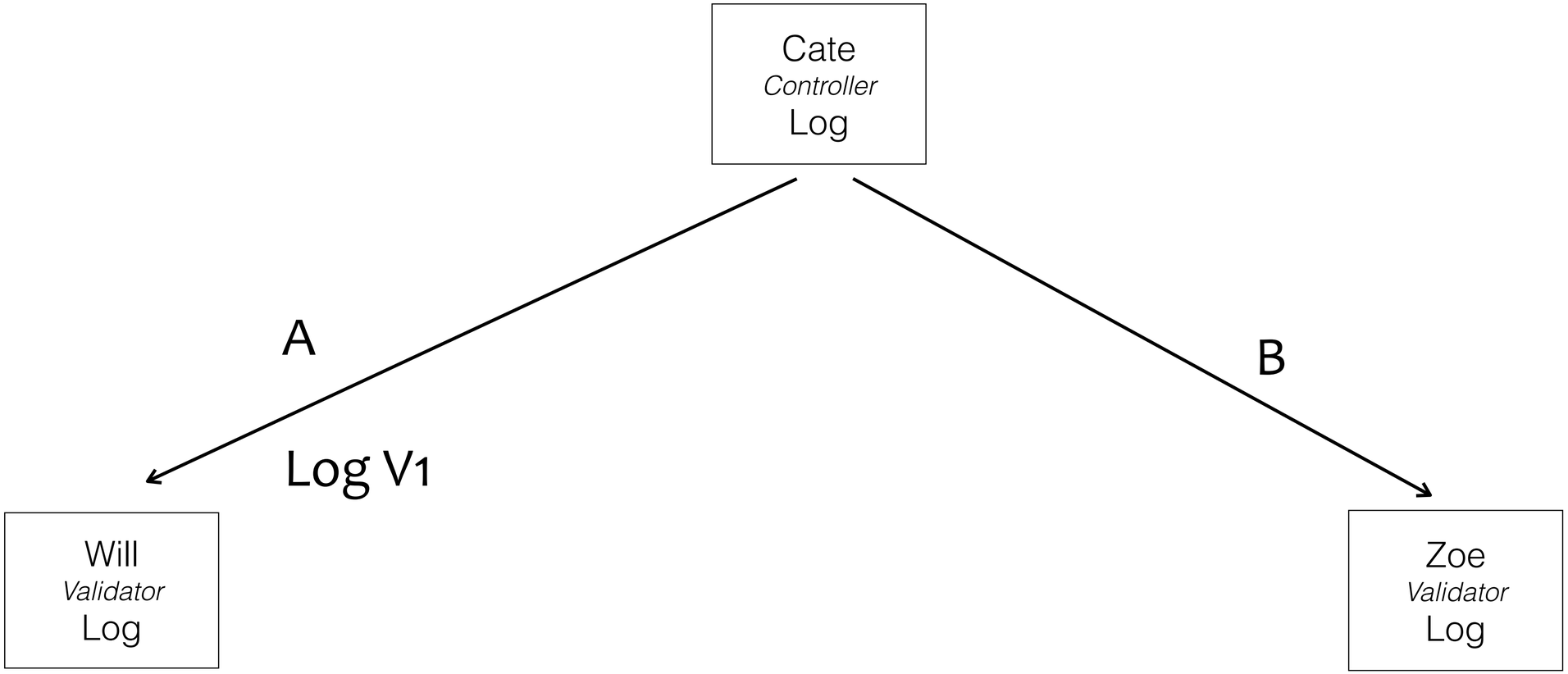
private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



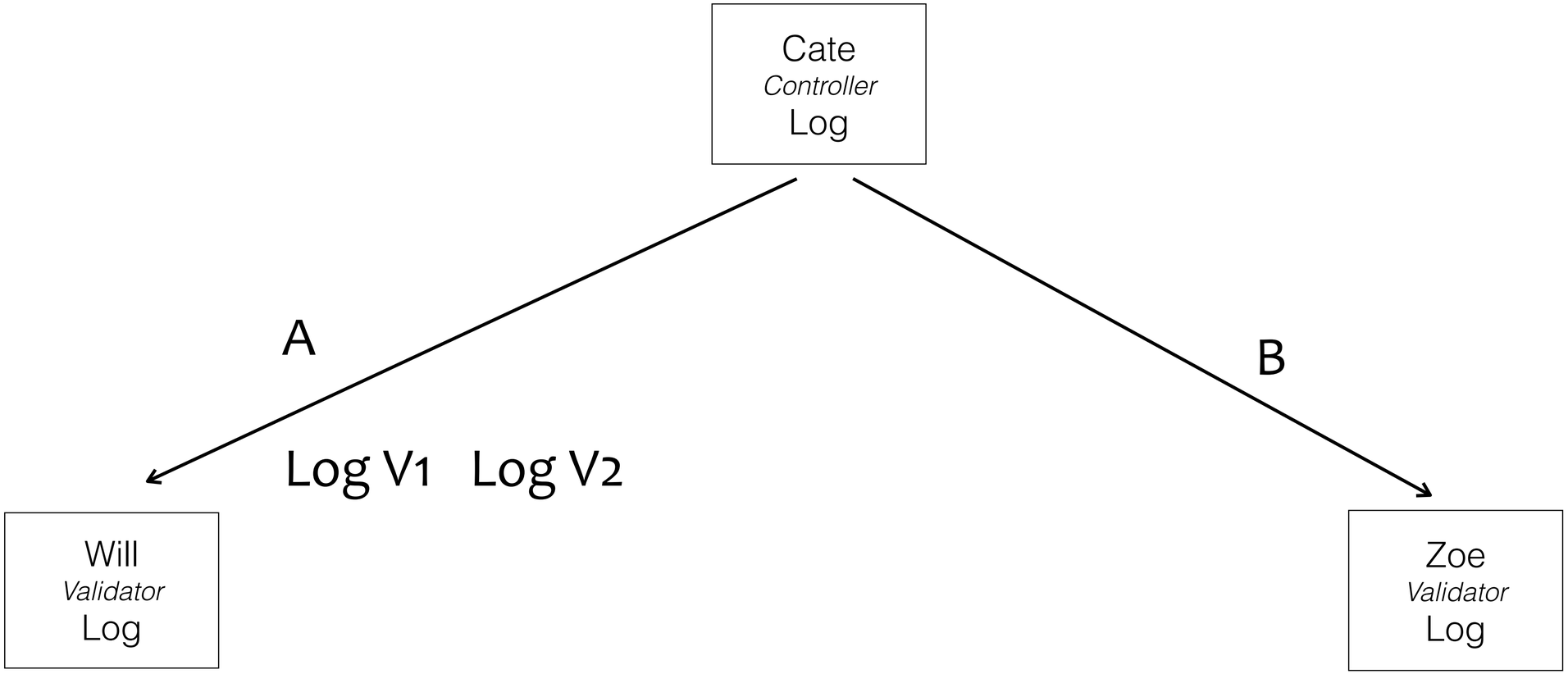
private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



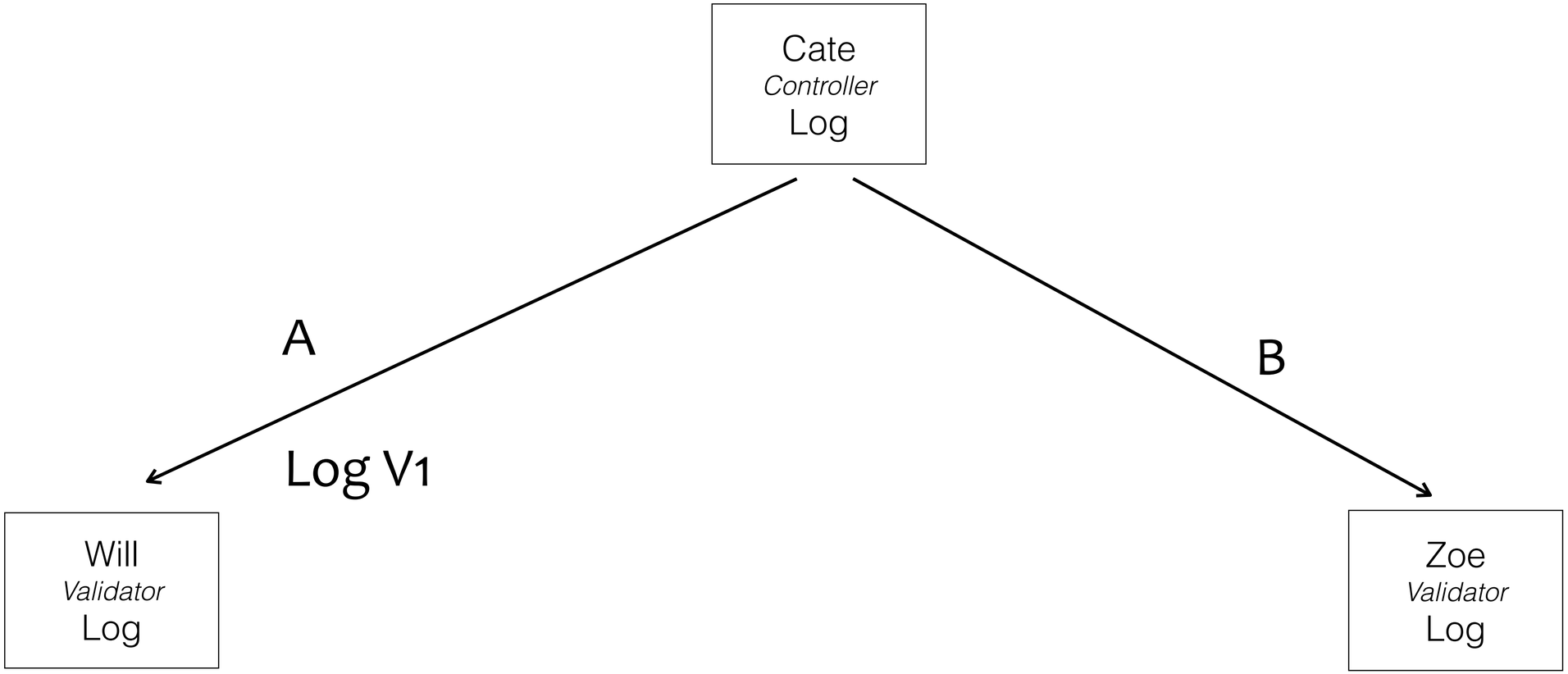
private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



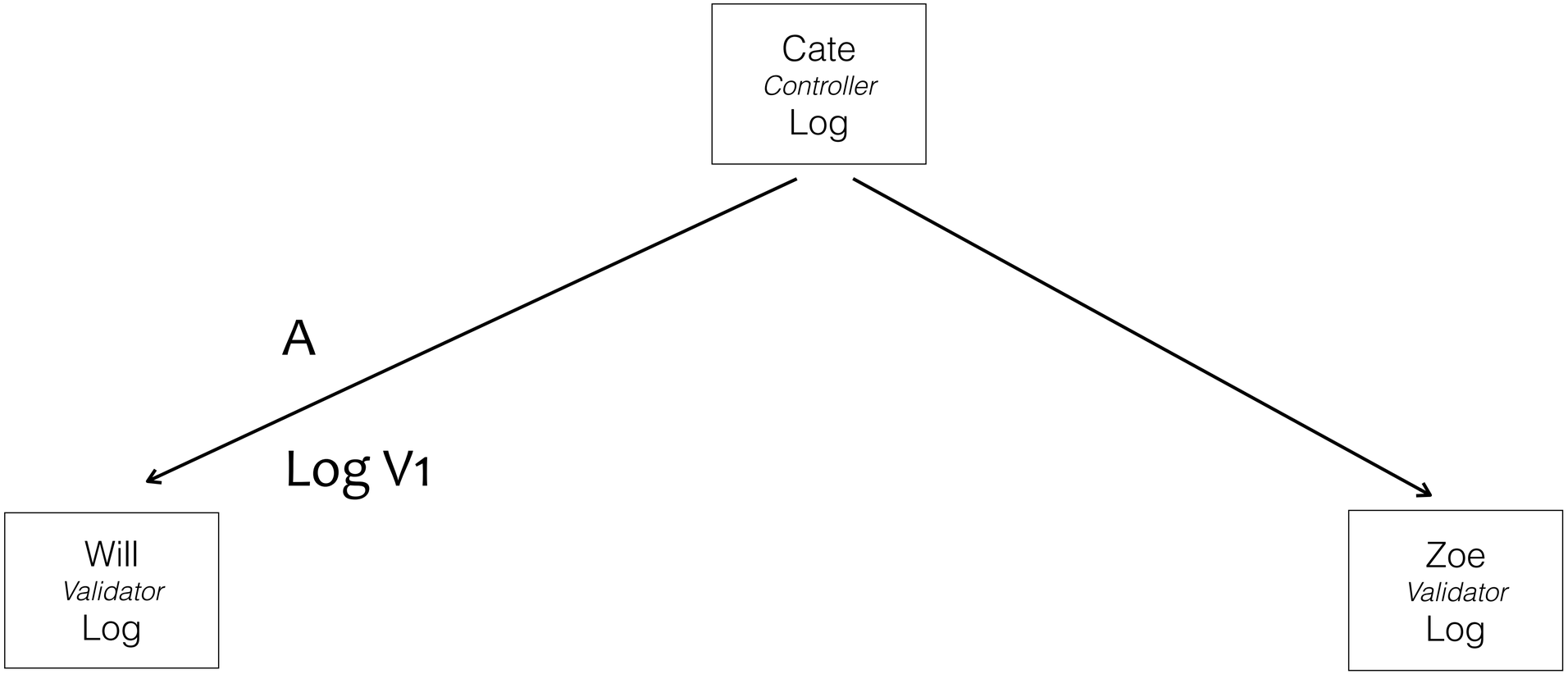
private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



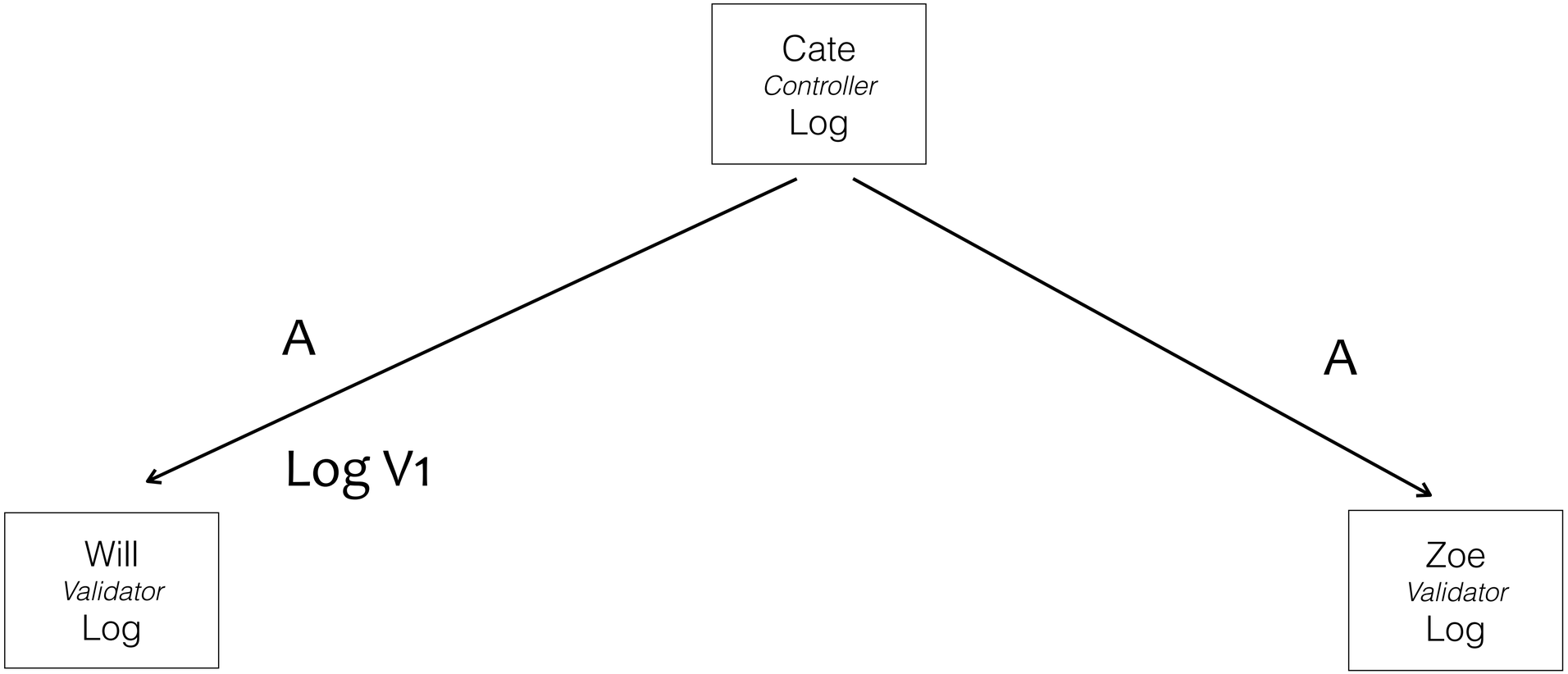
private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

Local Consistency Guarantee

How may Cate be duplicitous and not get caught?



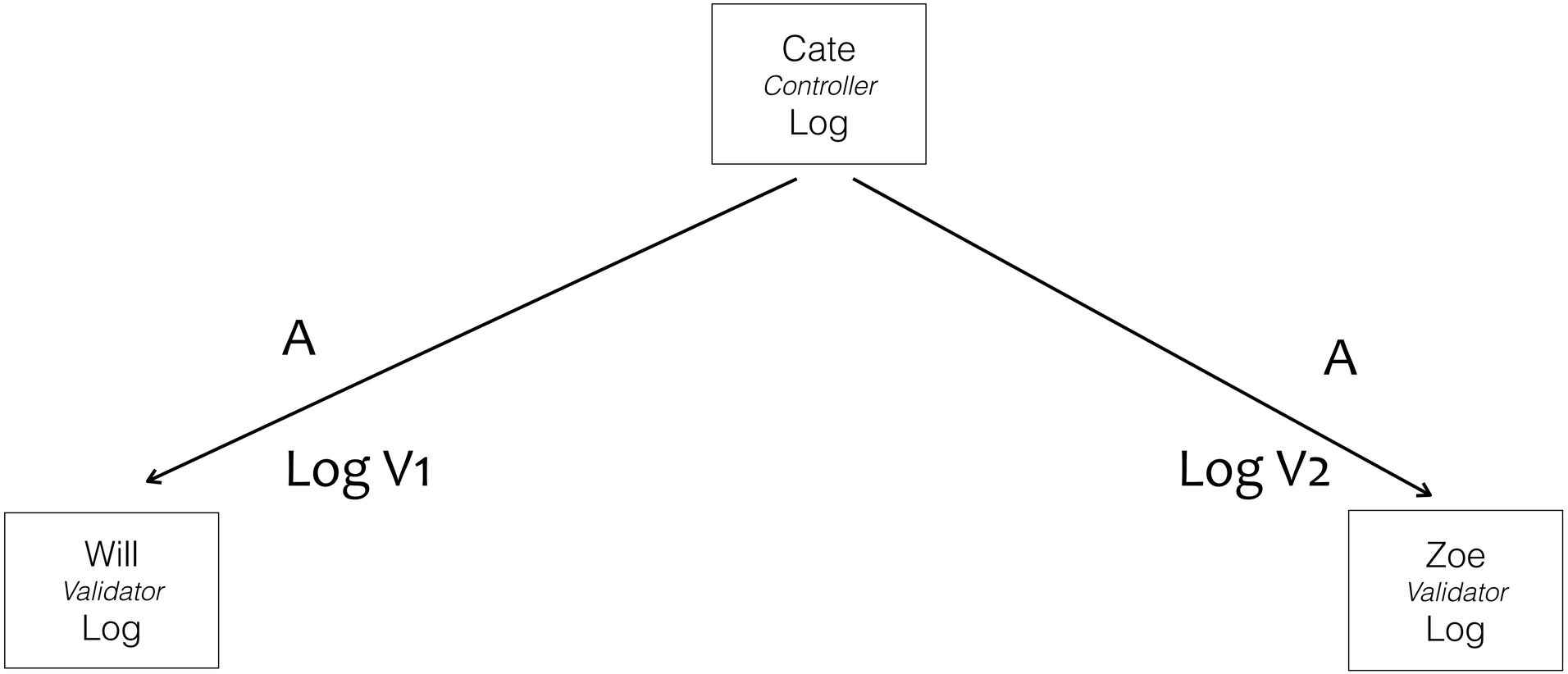
private (one-to-one) interactions

Duplicity Game

Cate promises to provide a consistent pair-wise log.

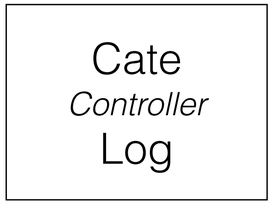
Local Consistency Guarantee

How may Cate be duplicitous and not get caught?

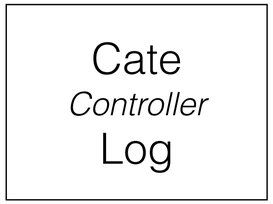


private (one-to-one) interactions

Duplicity Game

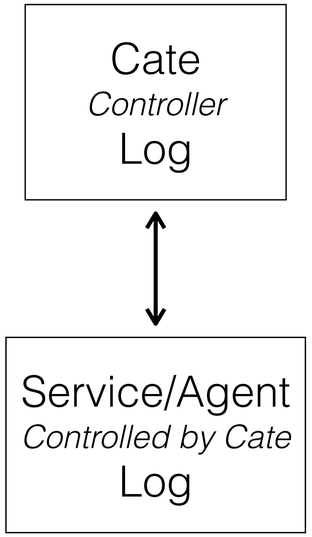


Duplicity Game



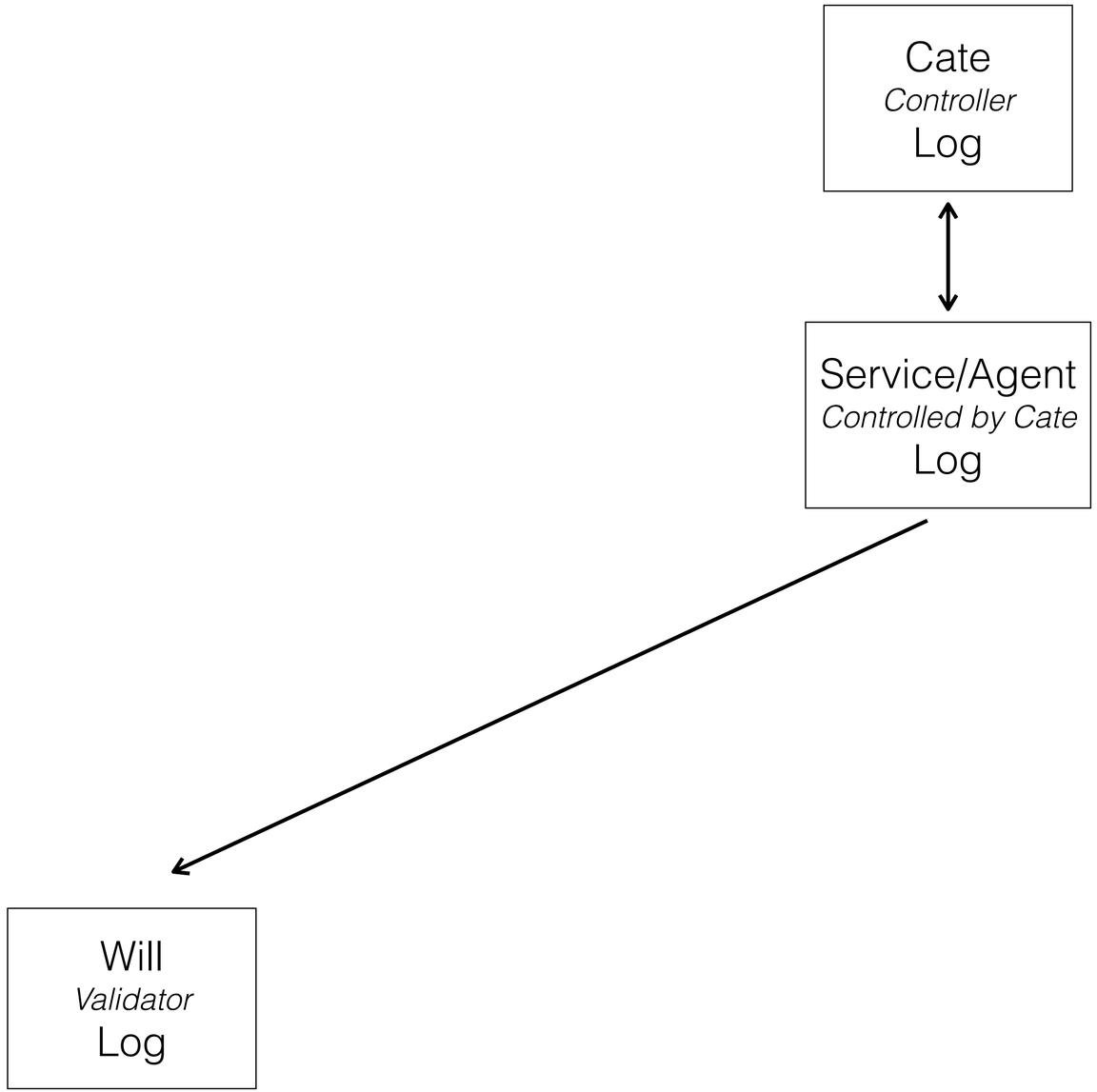
highly available, private (one-to-one) interactions

Duplicity Game



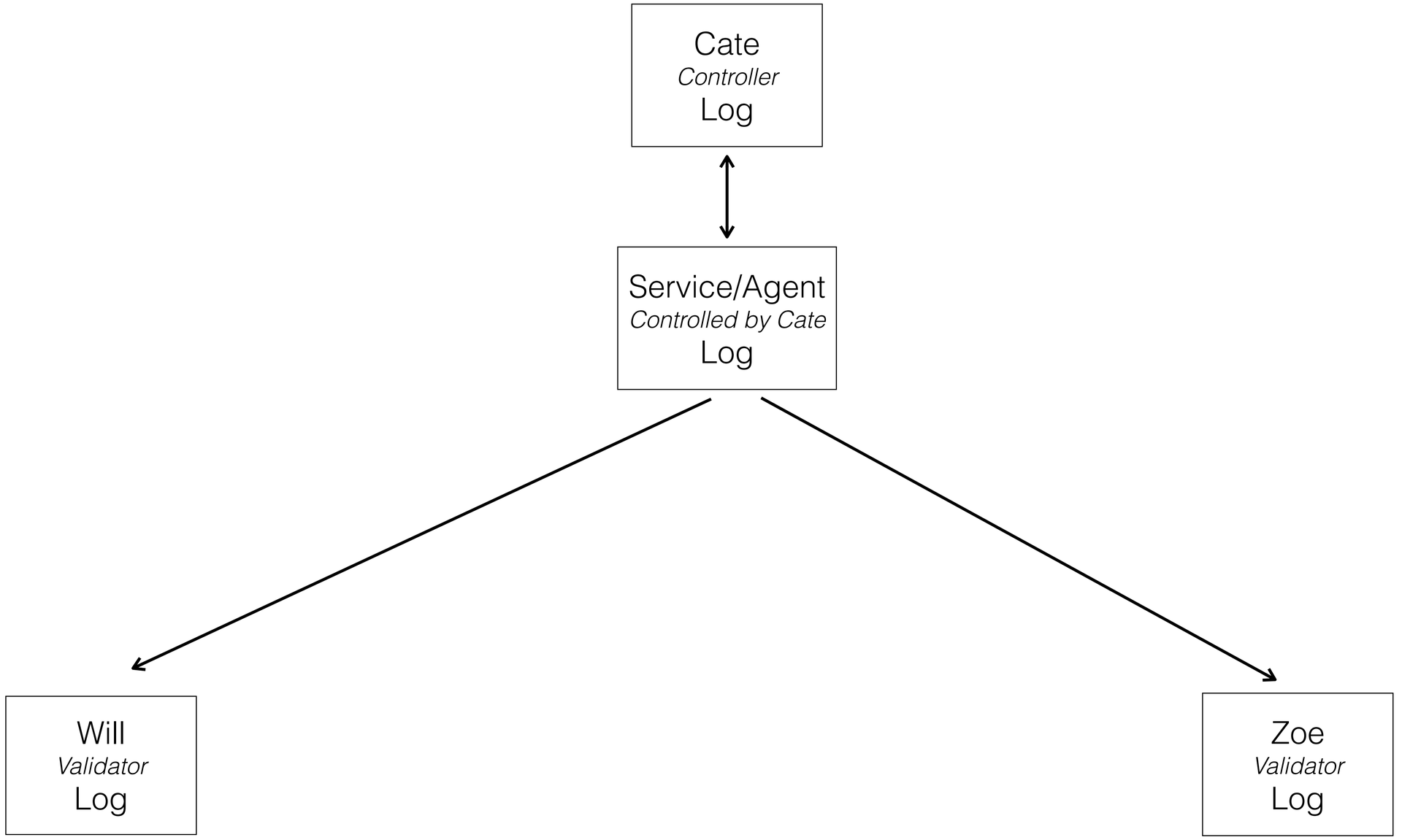
highly available, private (one-to-one) interactions

Duplicity Game



highly available, private (one-to-one) interactions

Duplicity Game

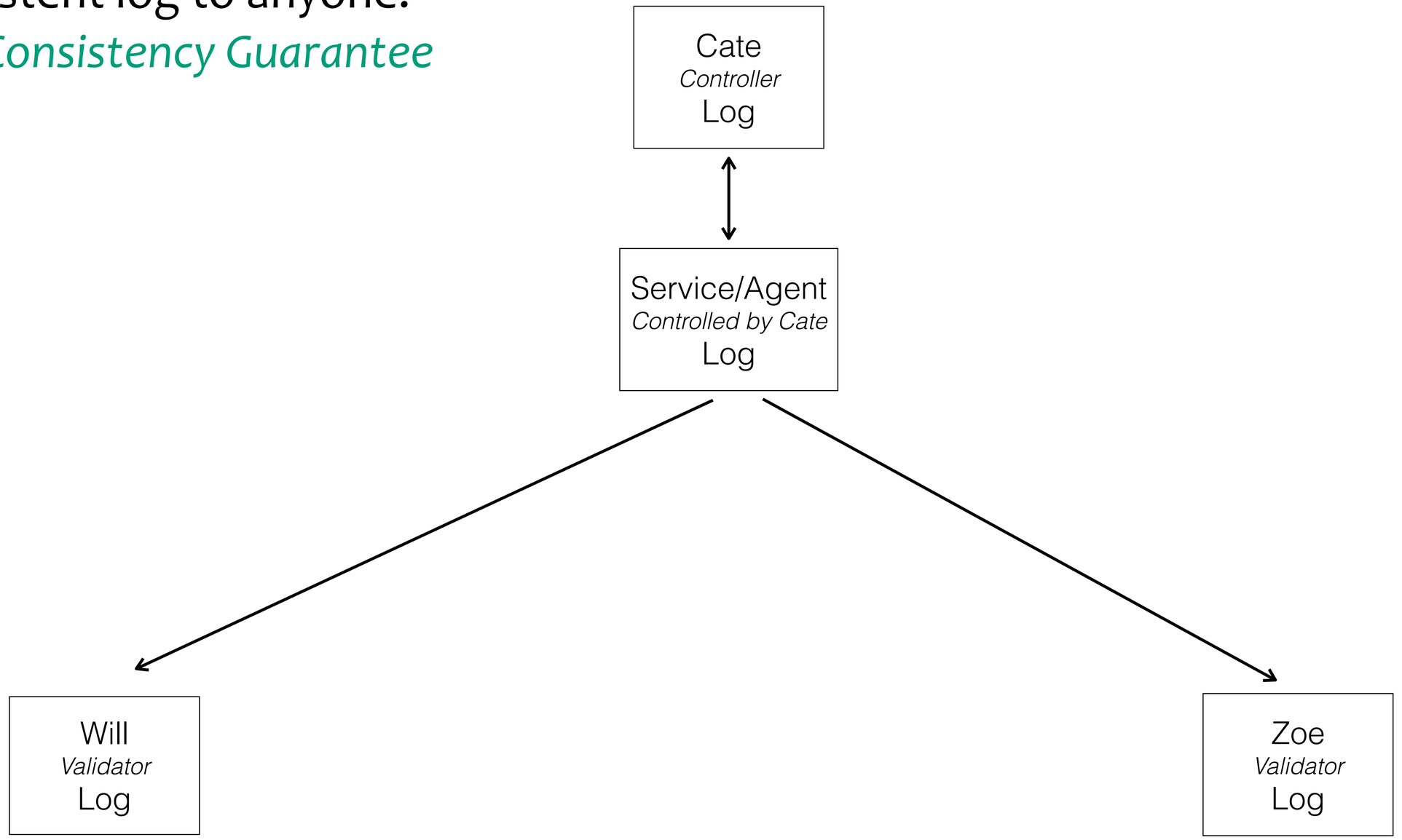


highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee



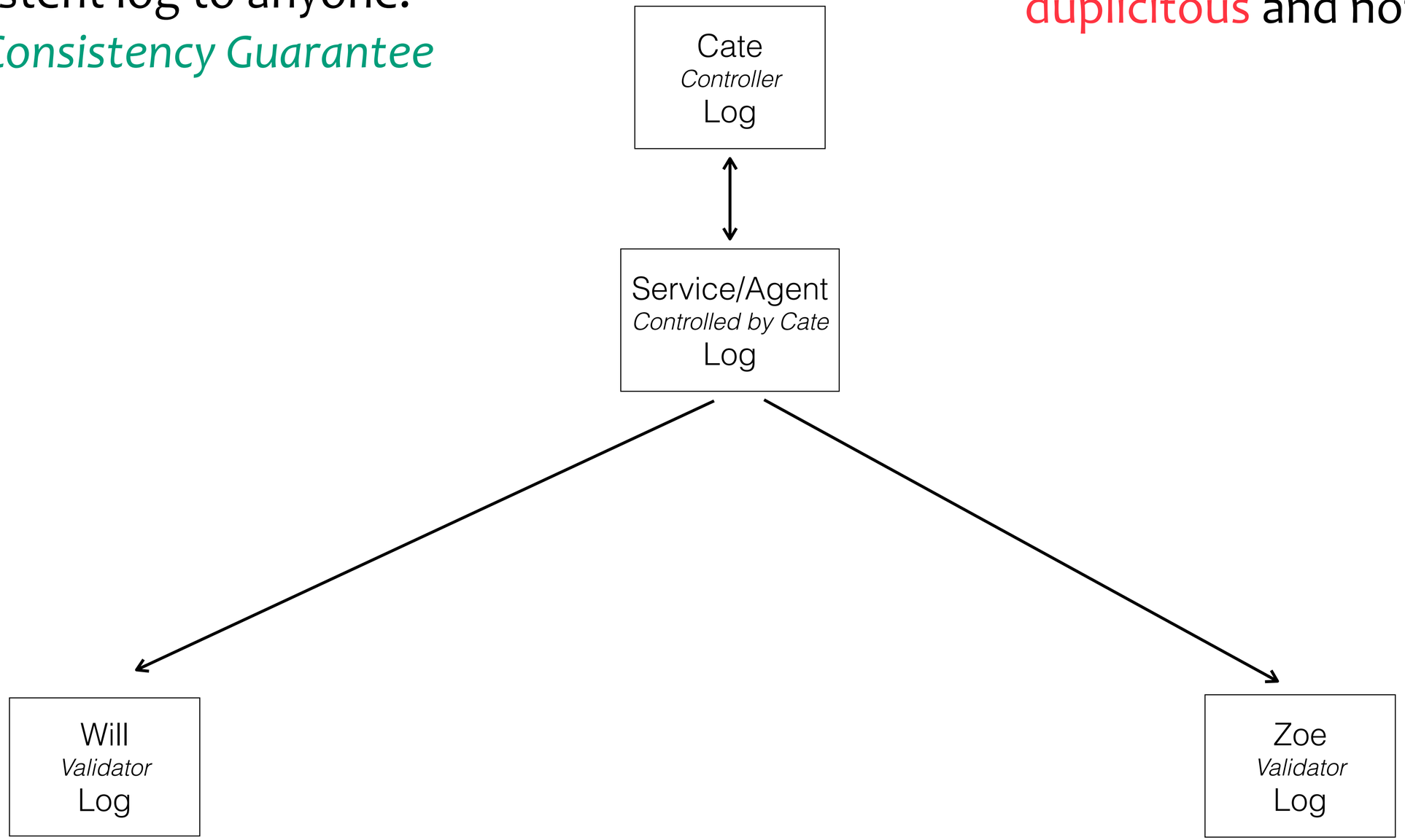
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



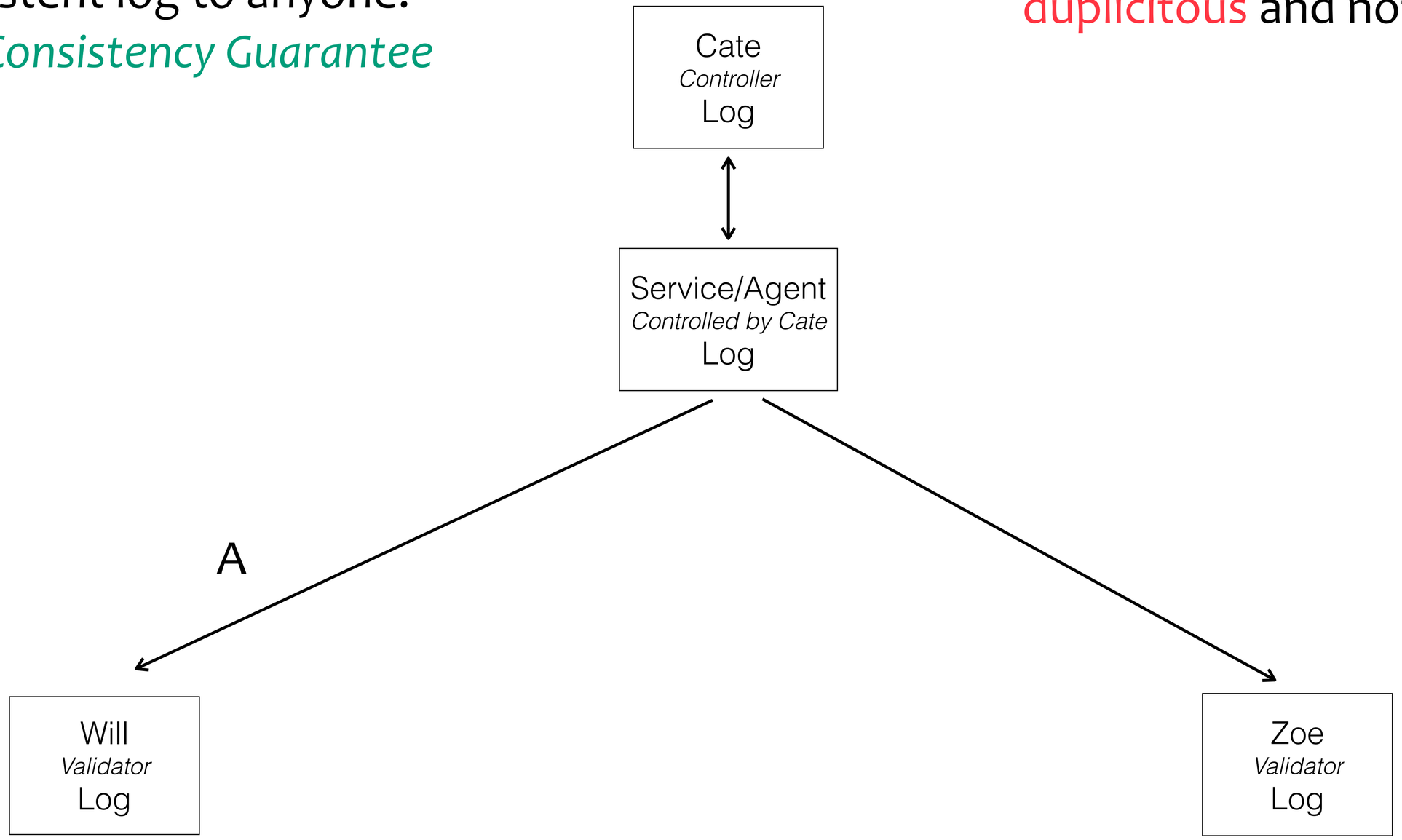
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



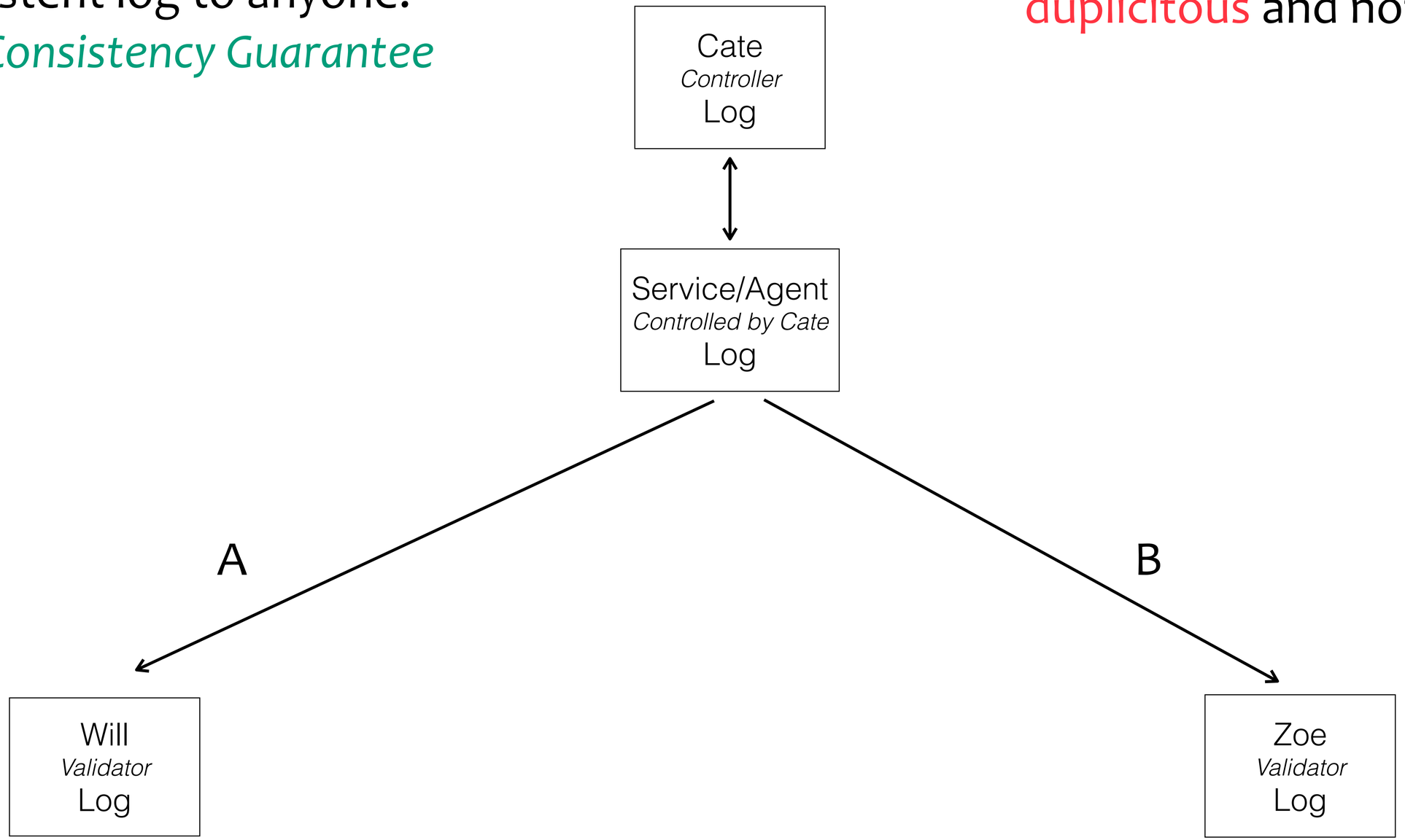
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



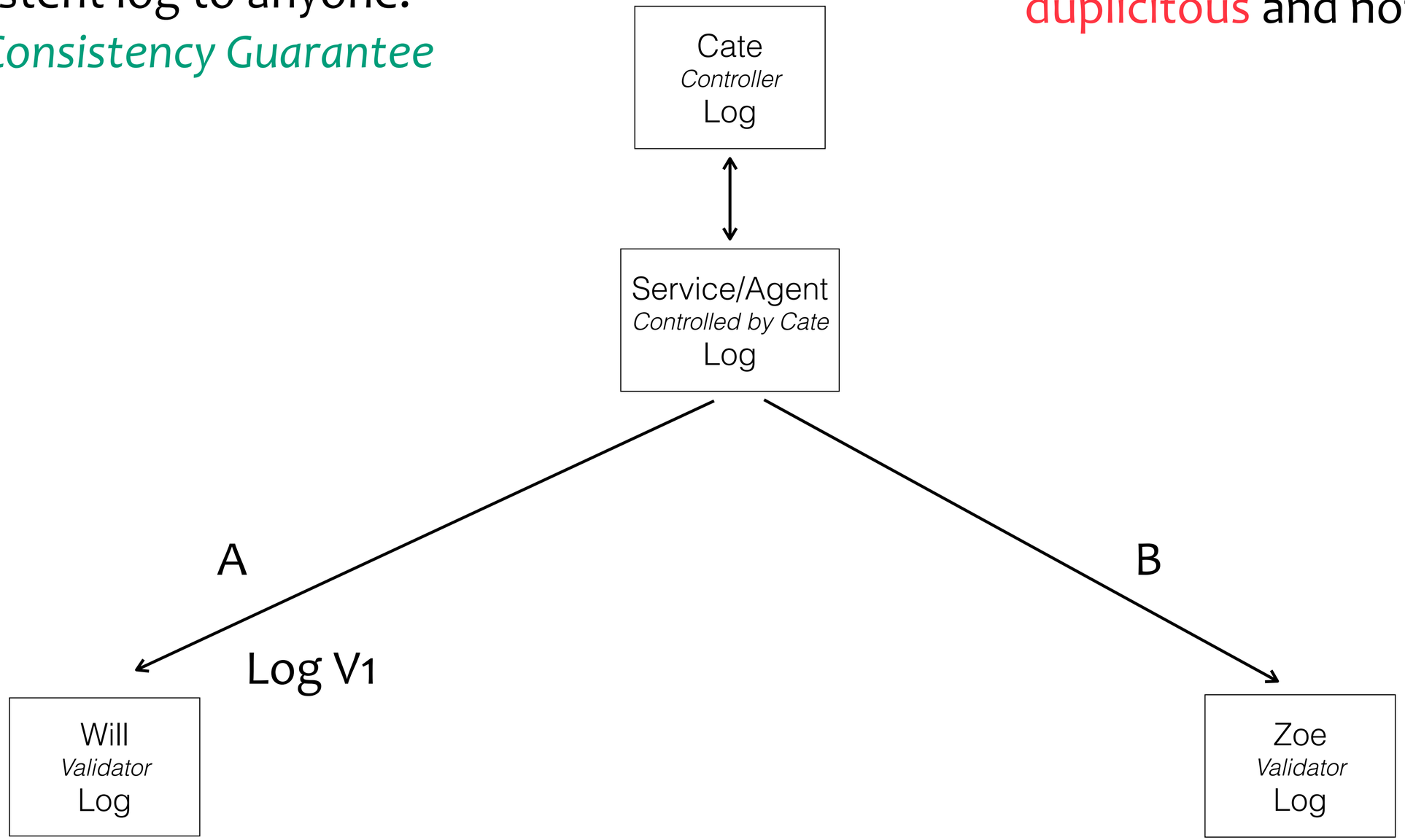
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



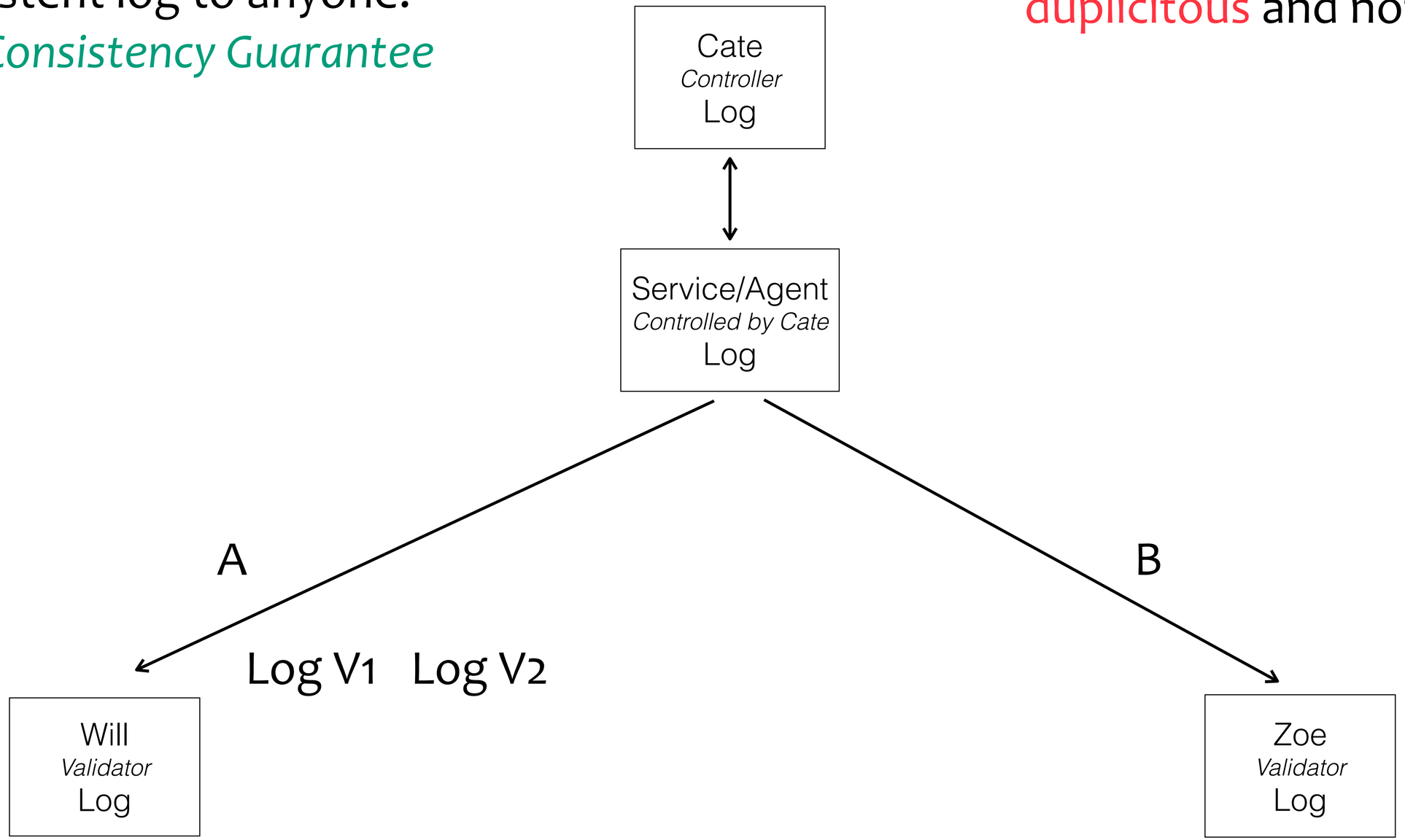
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



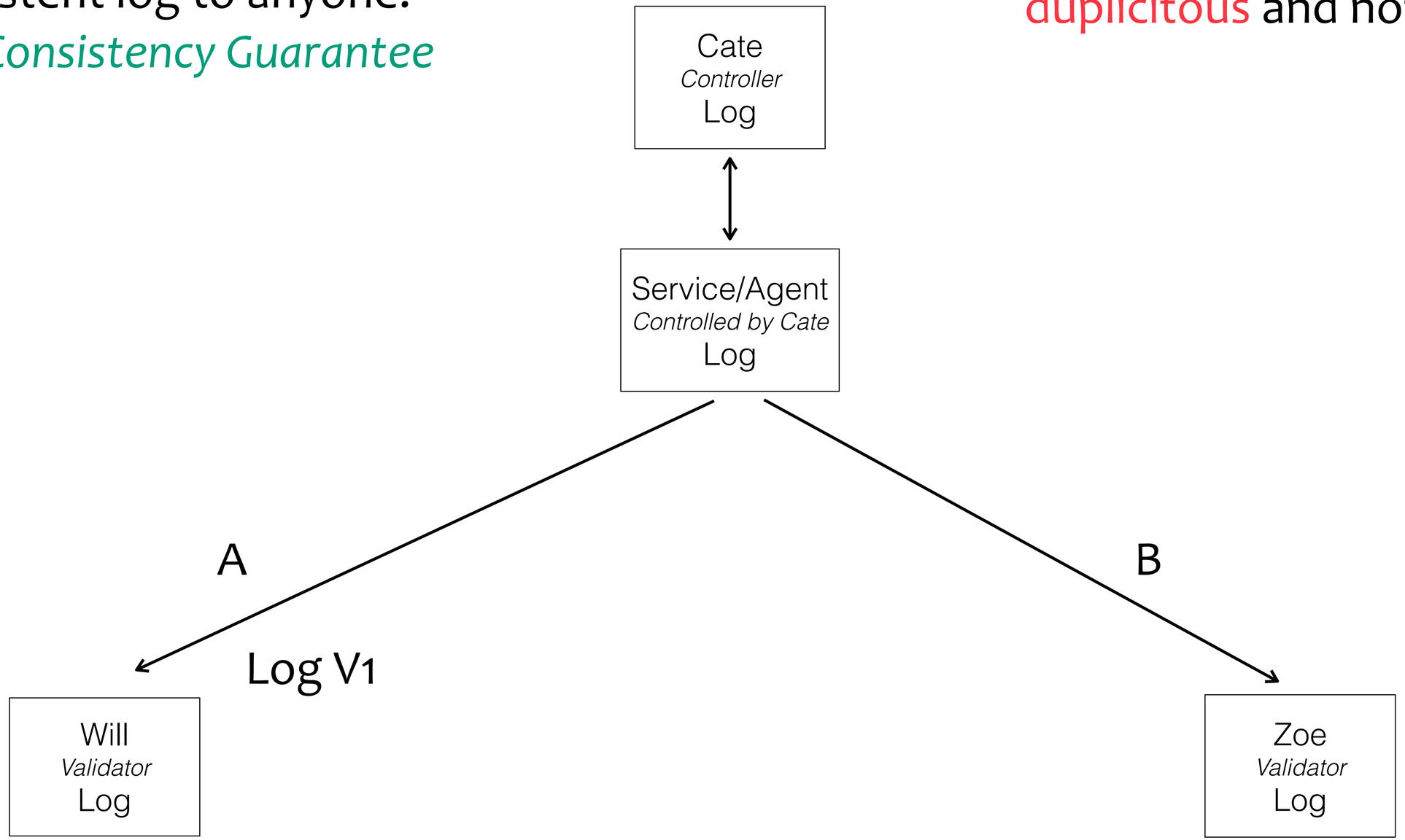
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



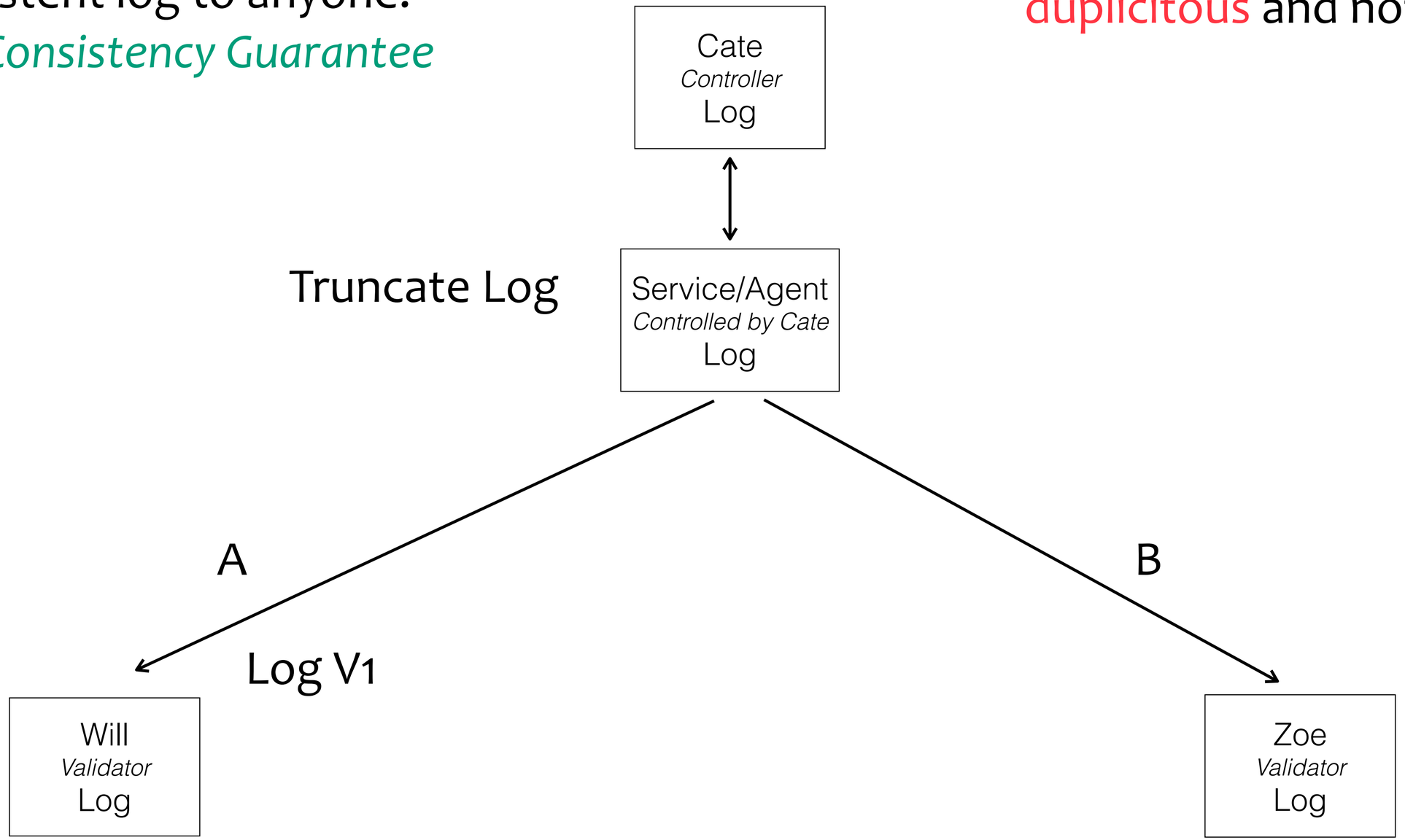
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



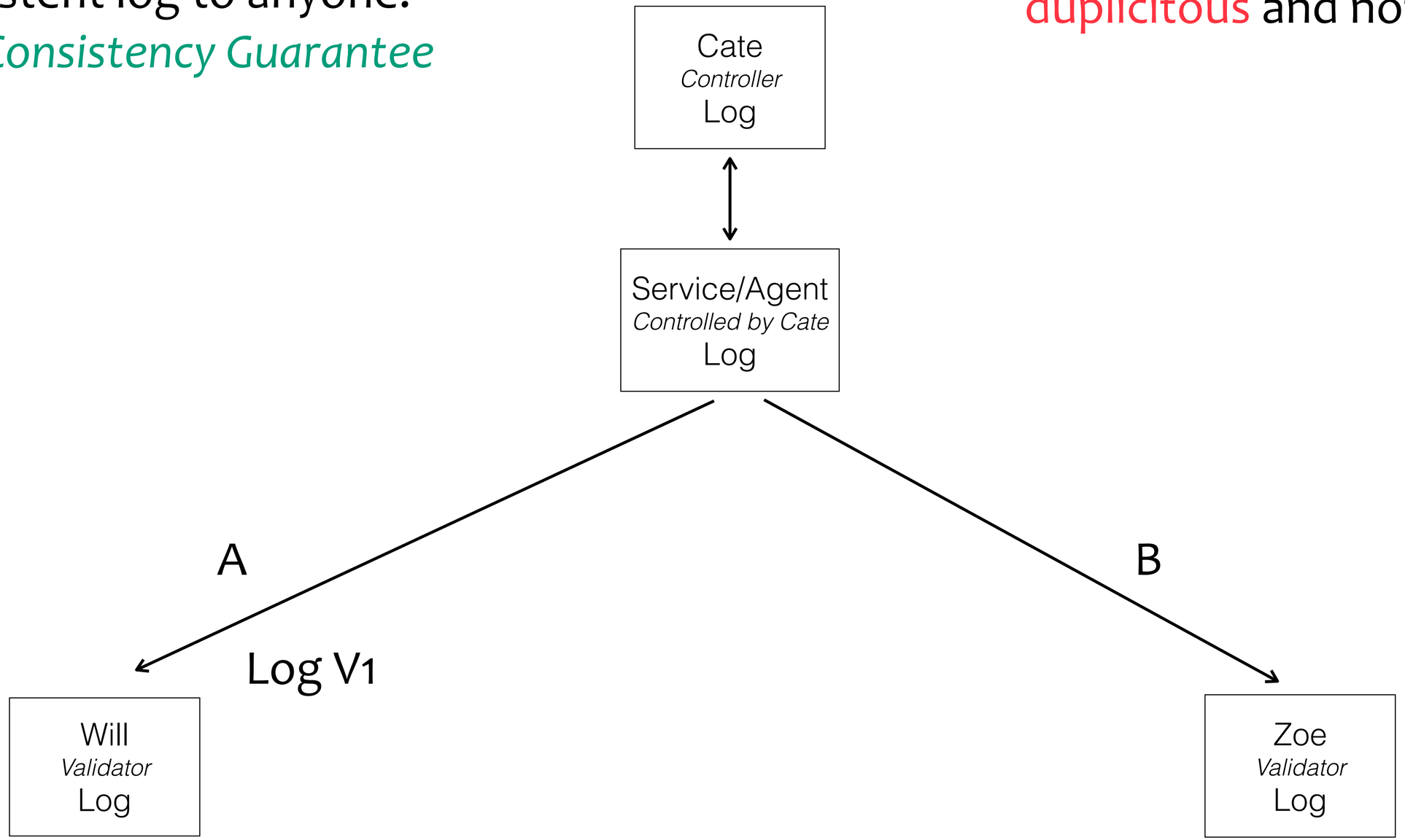
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



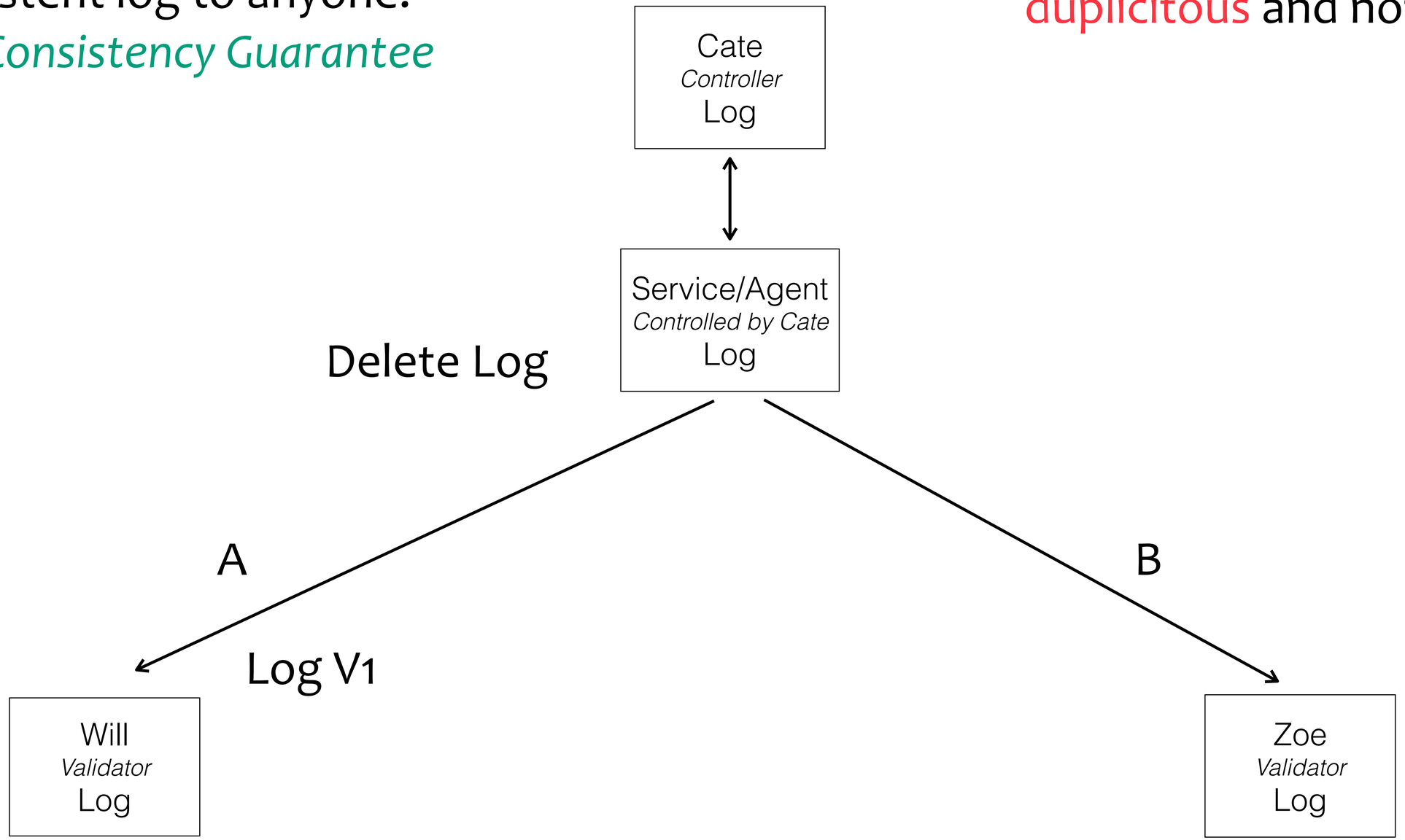
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



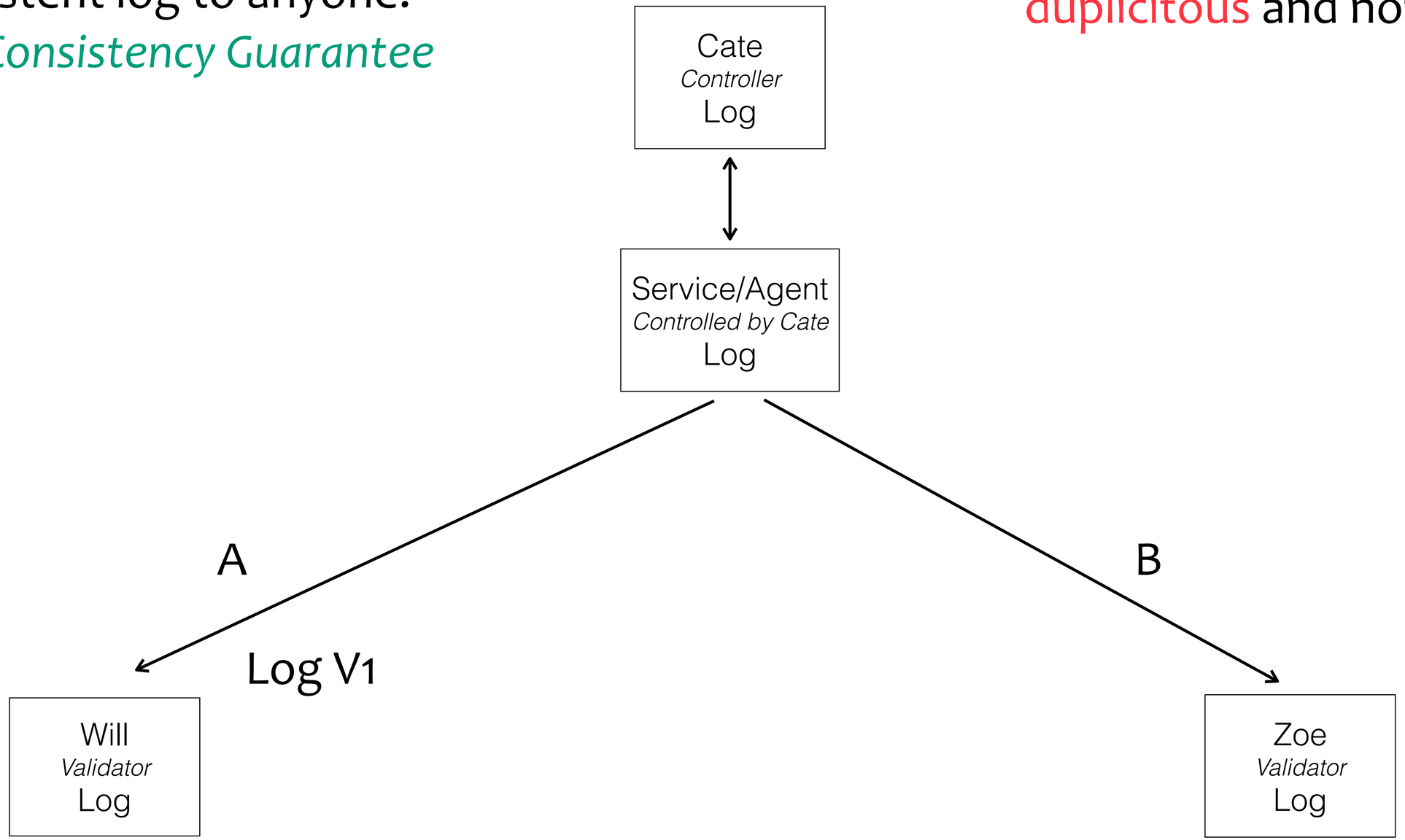
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



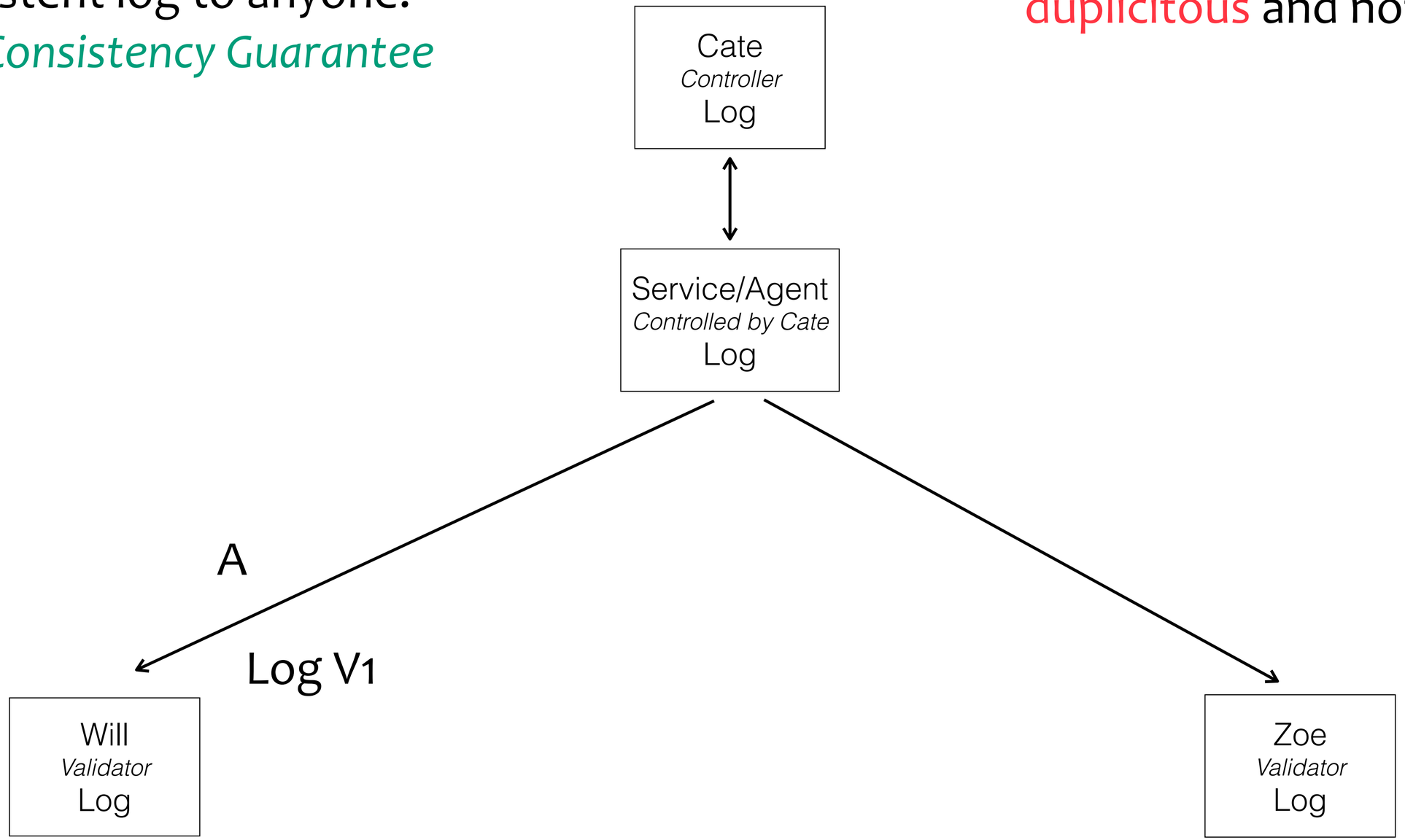
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



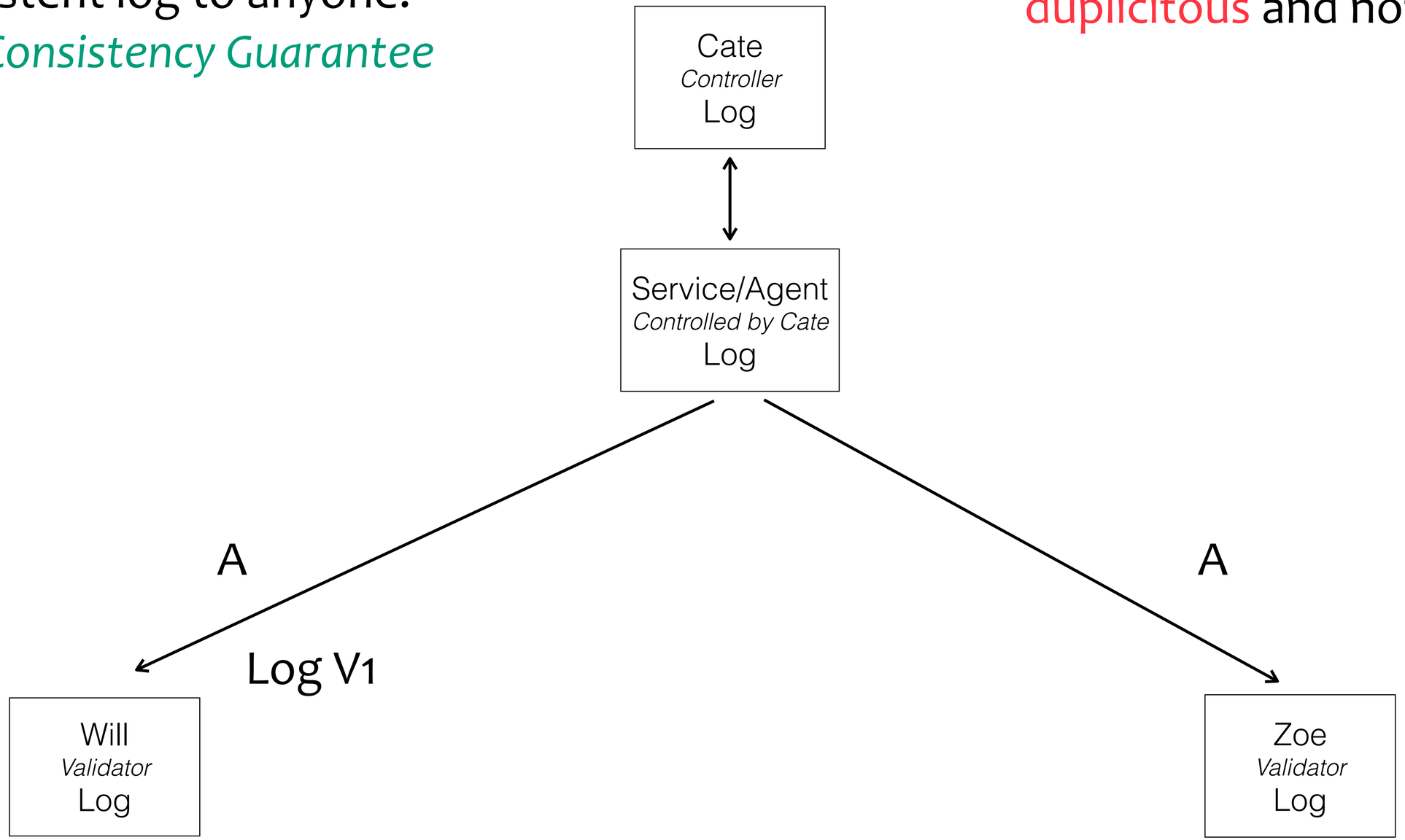
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?



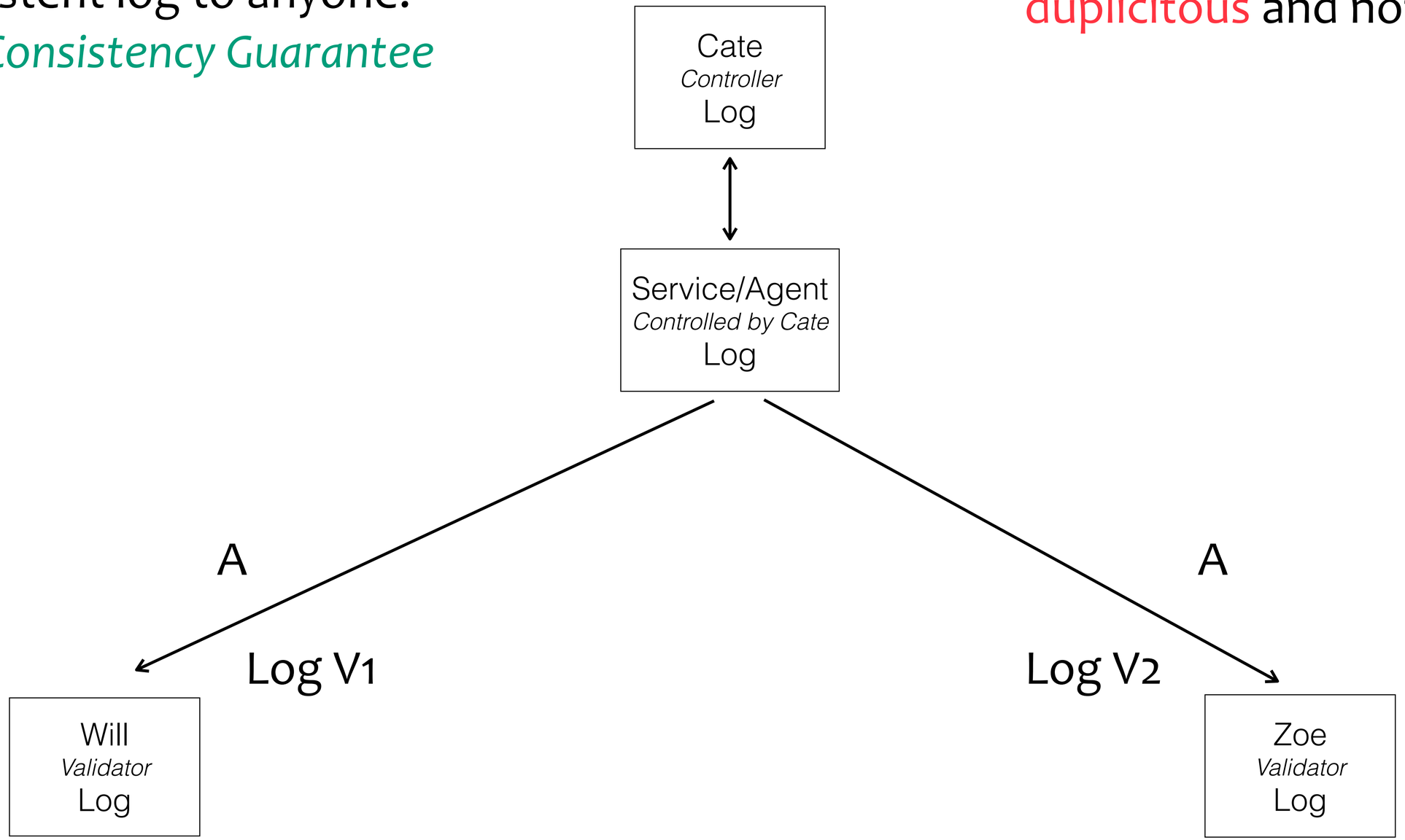
highly available, private (one-to-one) interactions

Duplicity Game

Service promises to provide a consistent log to anyone.

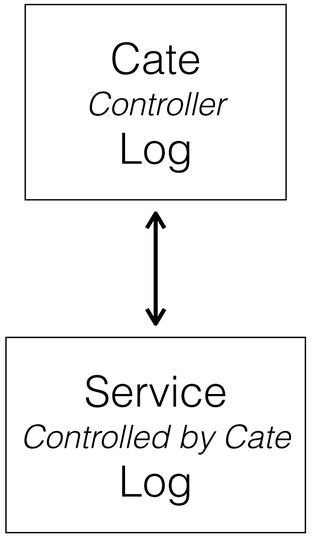
Local Consistency Guarantee

How may Cate/Service/Agent be duplicitous and not get caught?

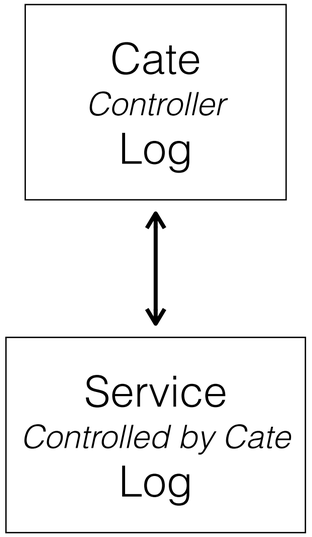


highly available, private (one-to-one) interactions

Duplicity Game

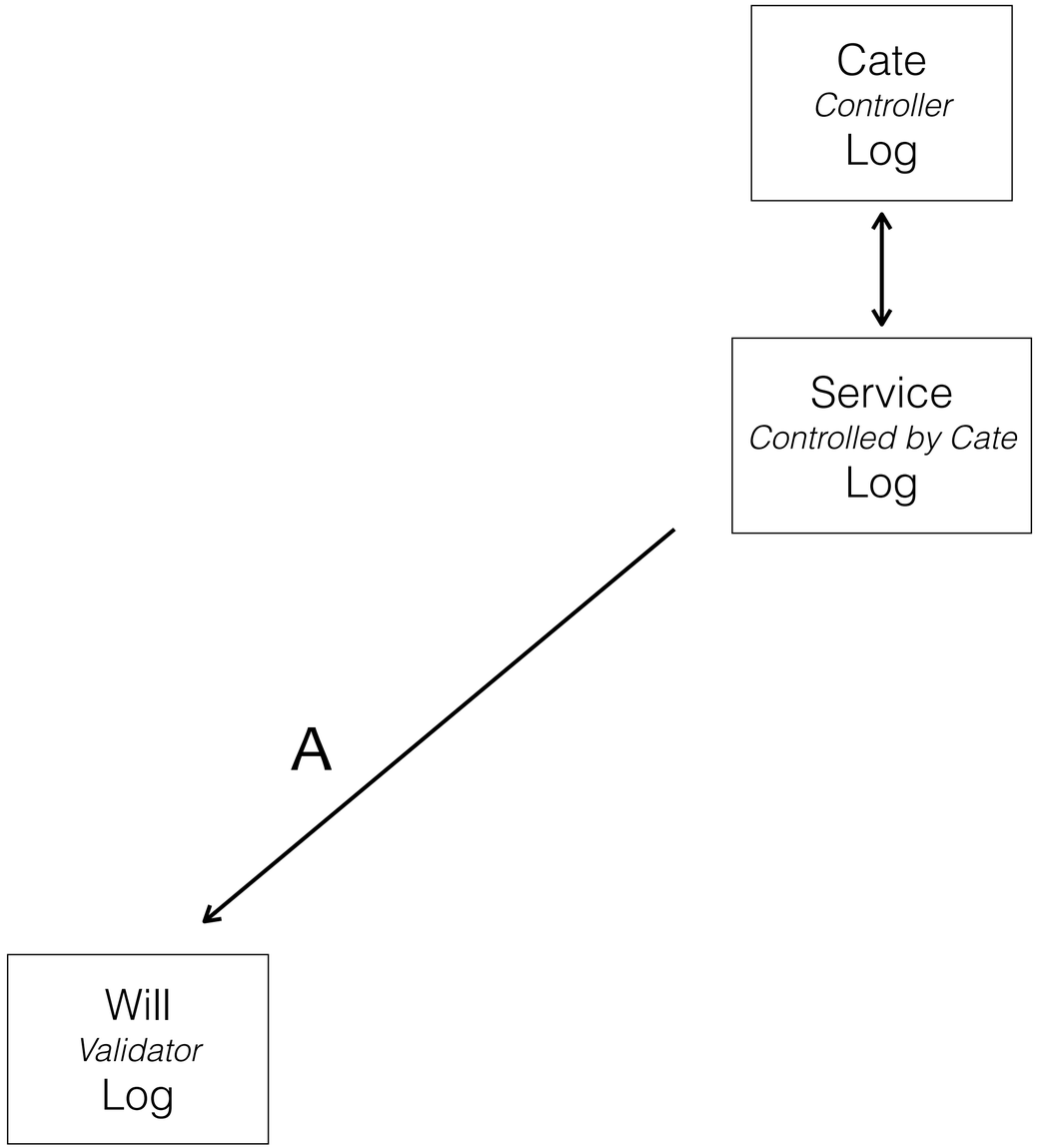


Duplicity Game



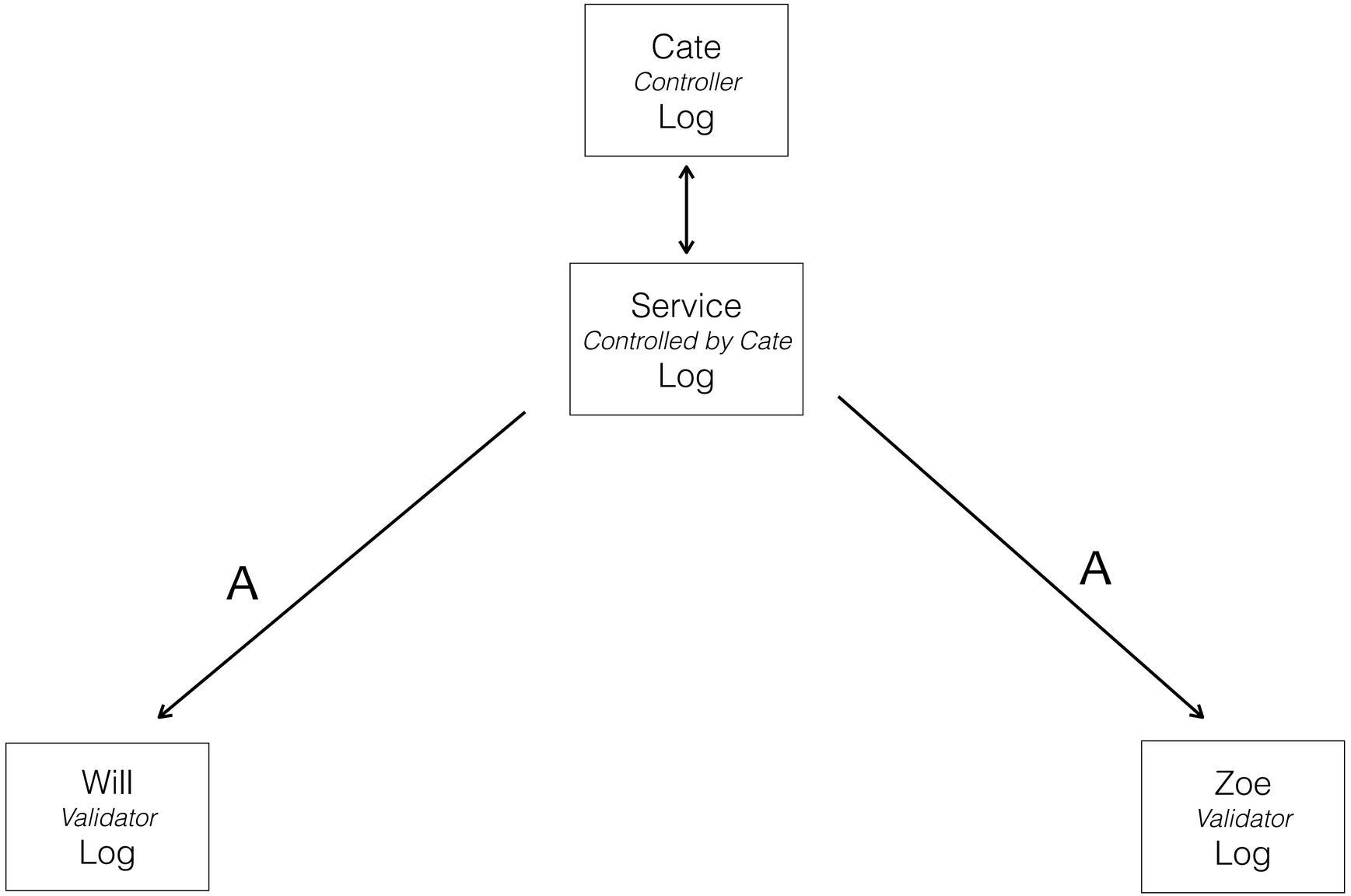
global consistent, highly available, and public (one-to-any) interactions

Duplicity Game



global consistent, highly available, and public (one-to-any) interactions

Duplicity Game

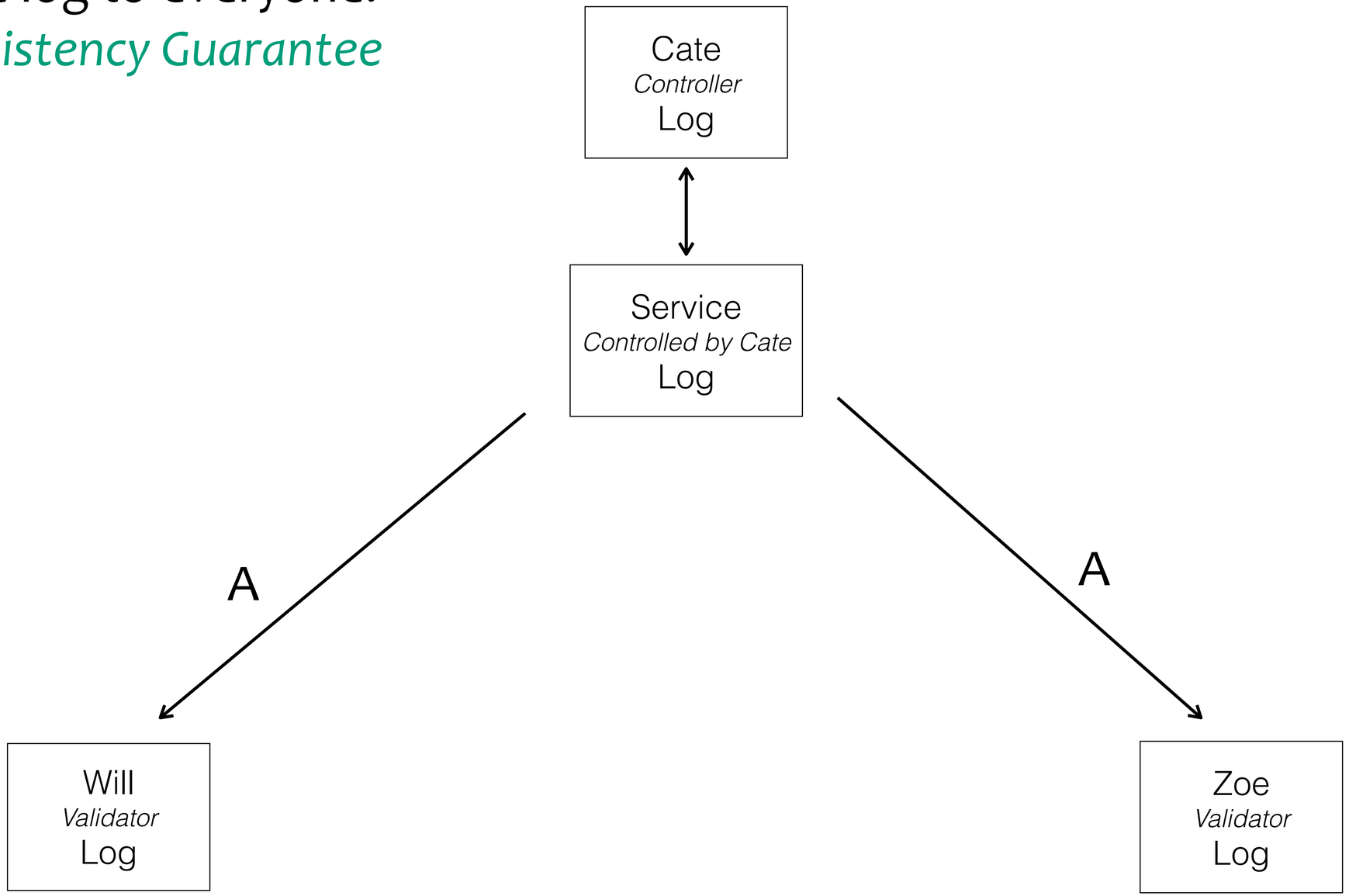


global consistent, highly available, and public (one-to-any) interactions

Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee



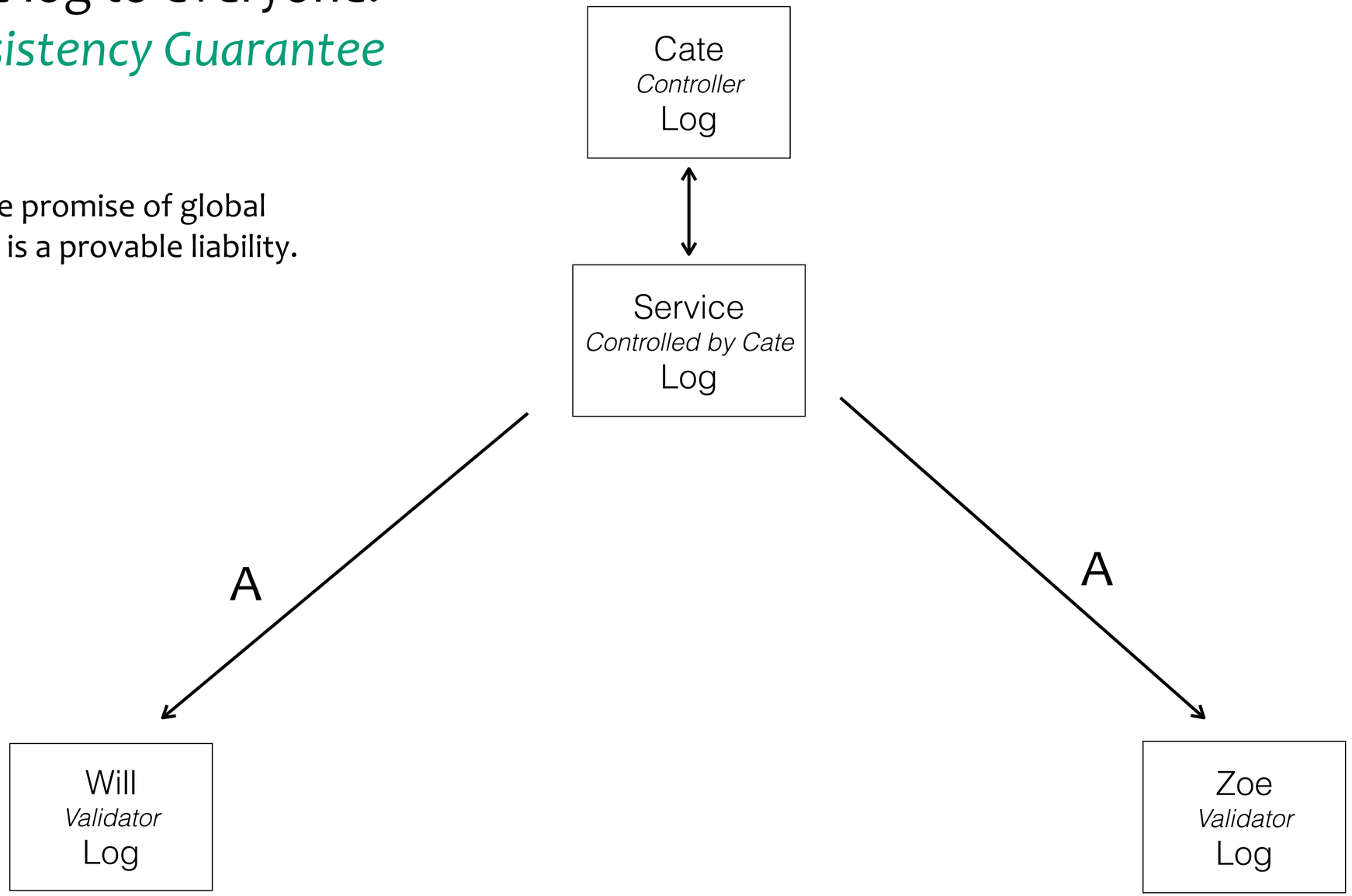
global consistent, highly available, and public (one-to-any) interactions

Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

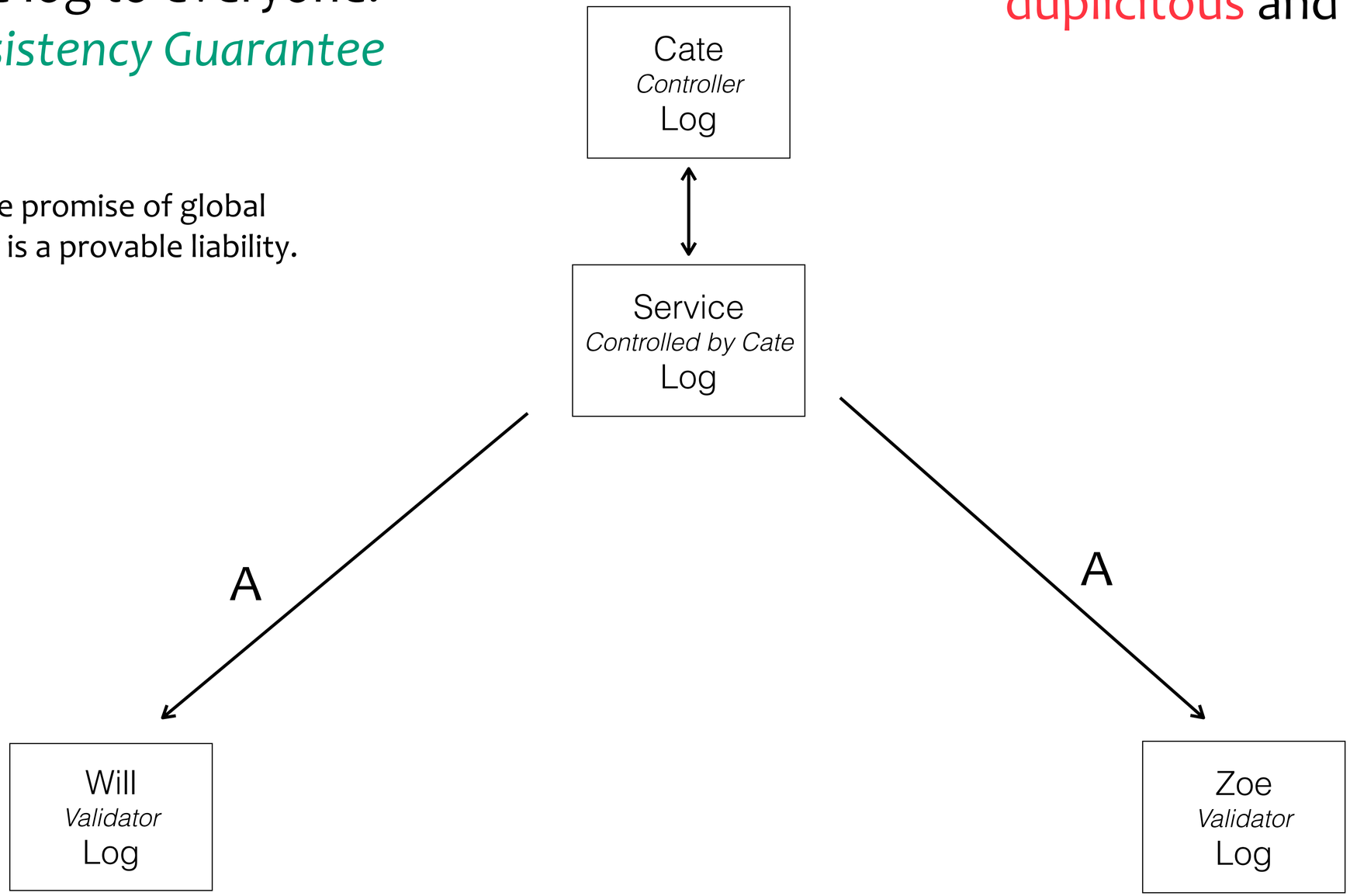
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

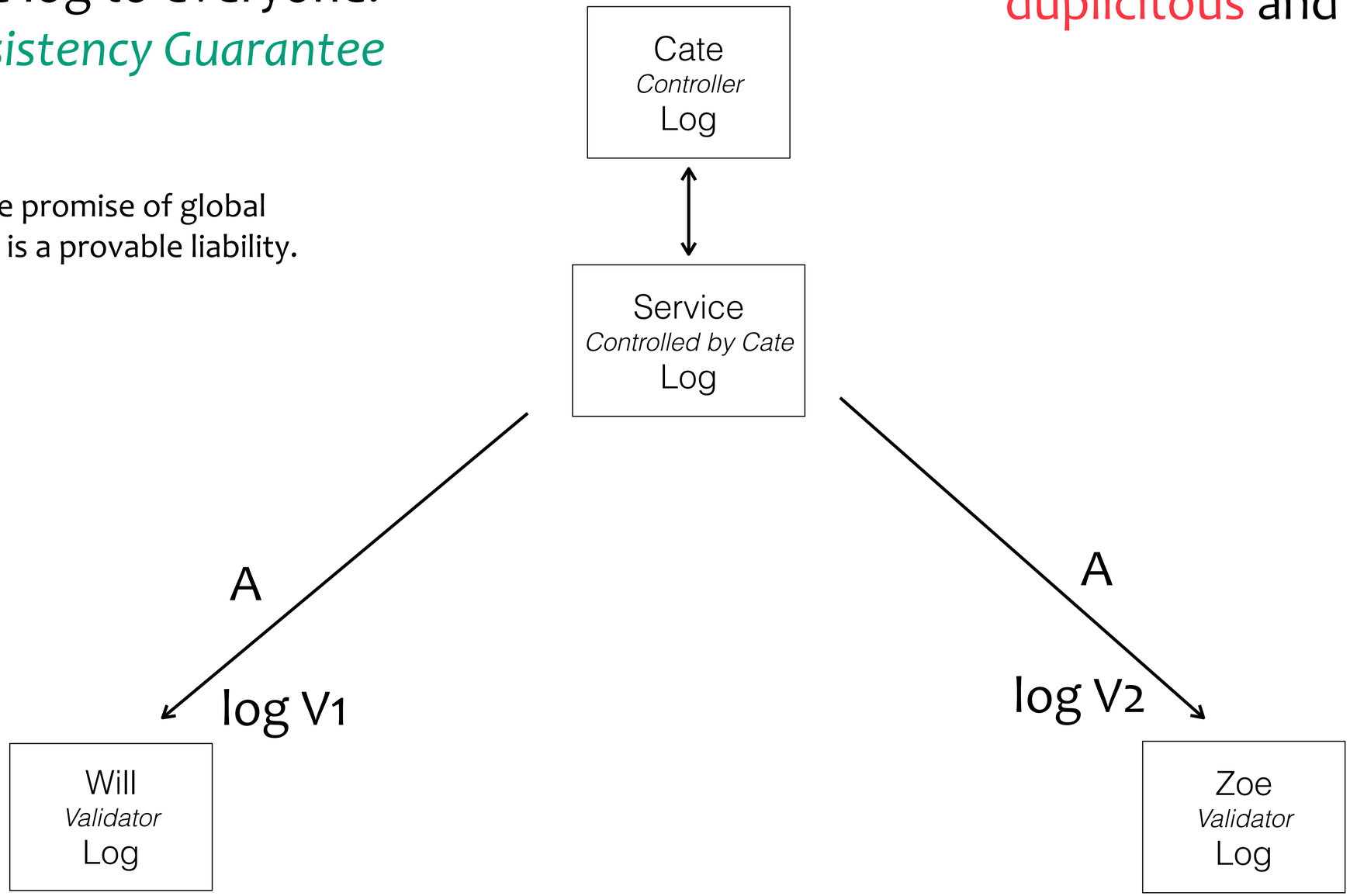
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

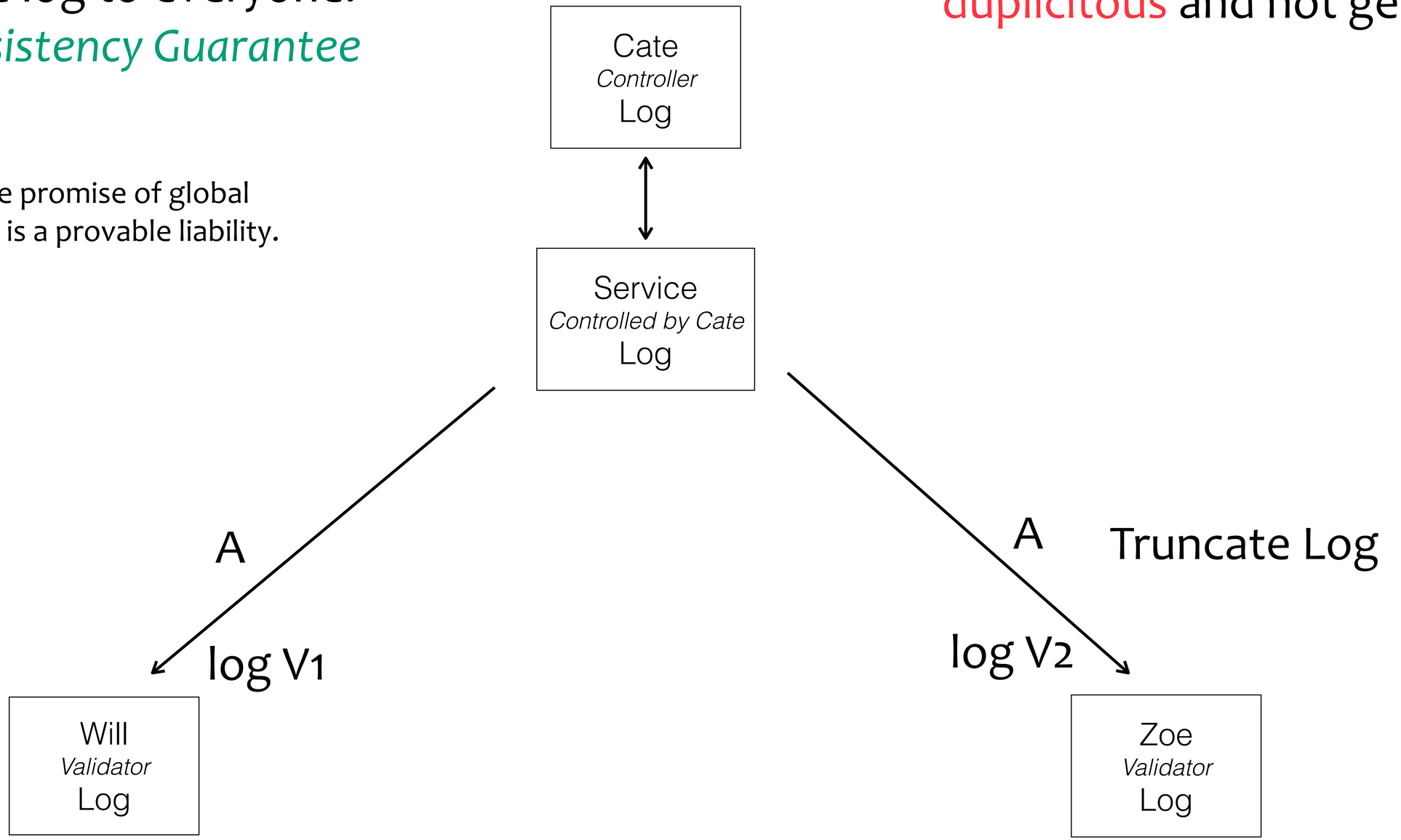
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

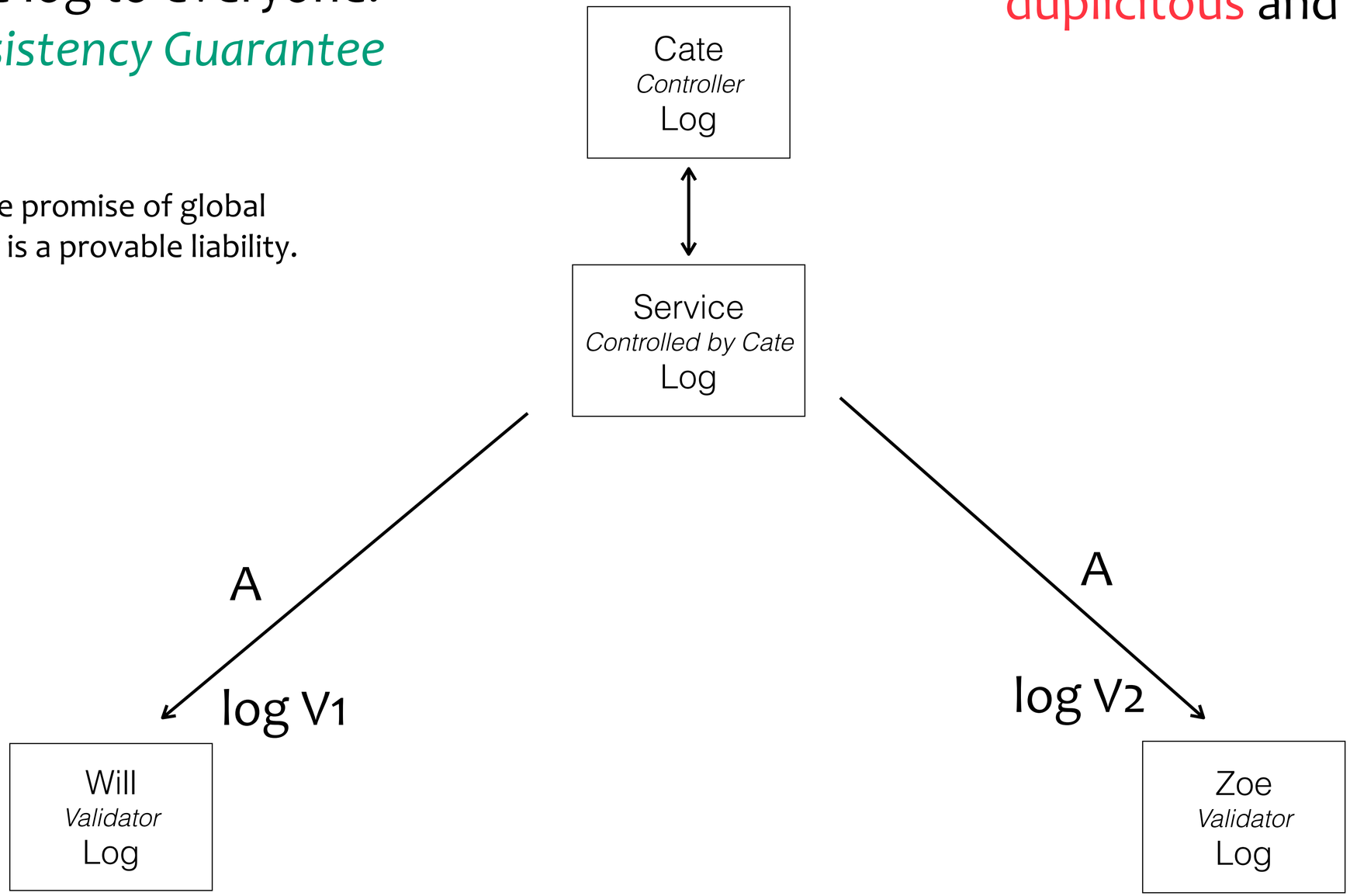
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

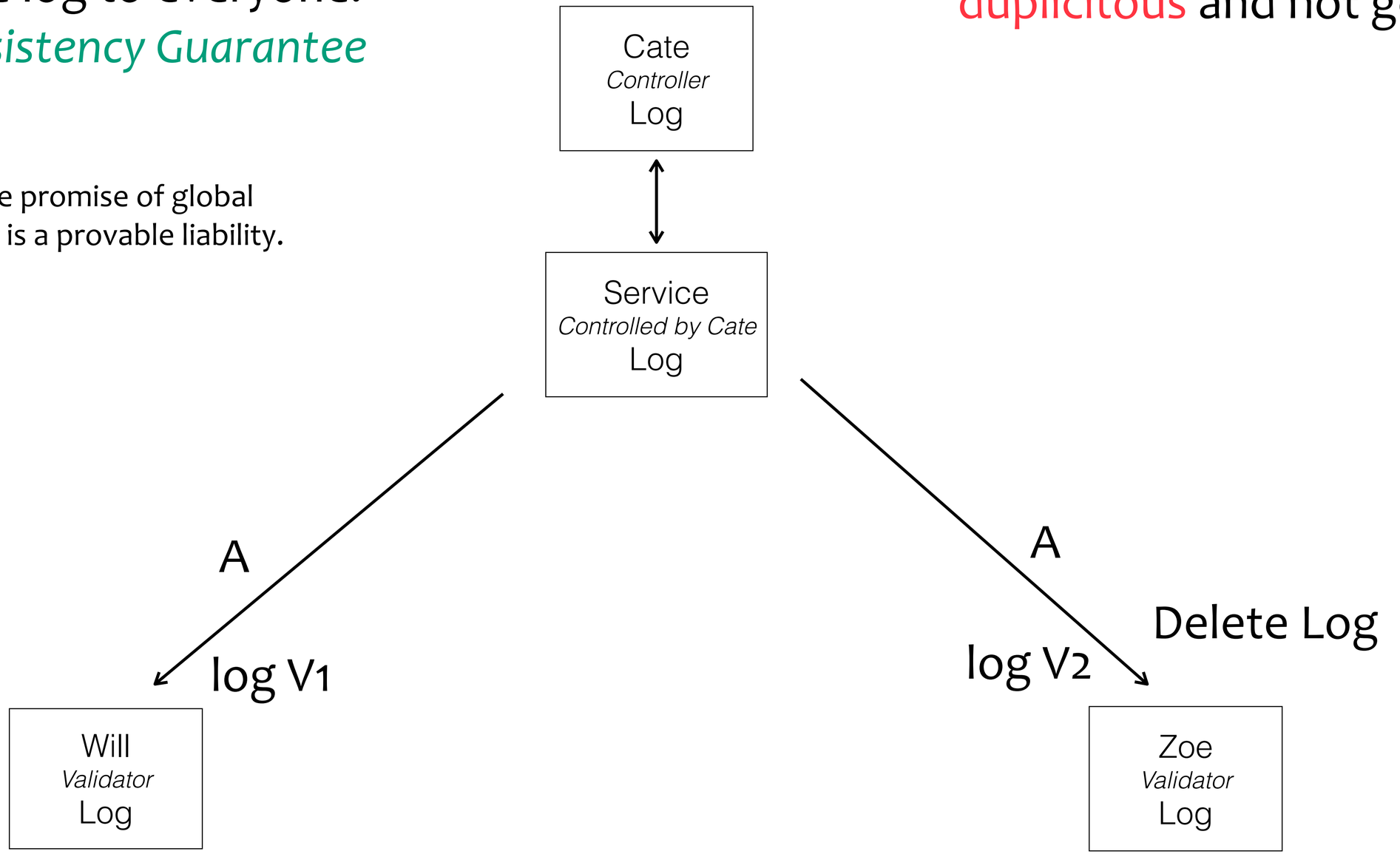
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

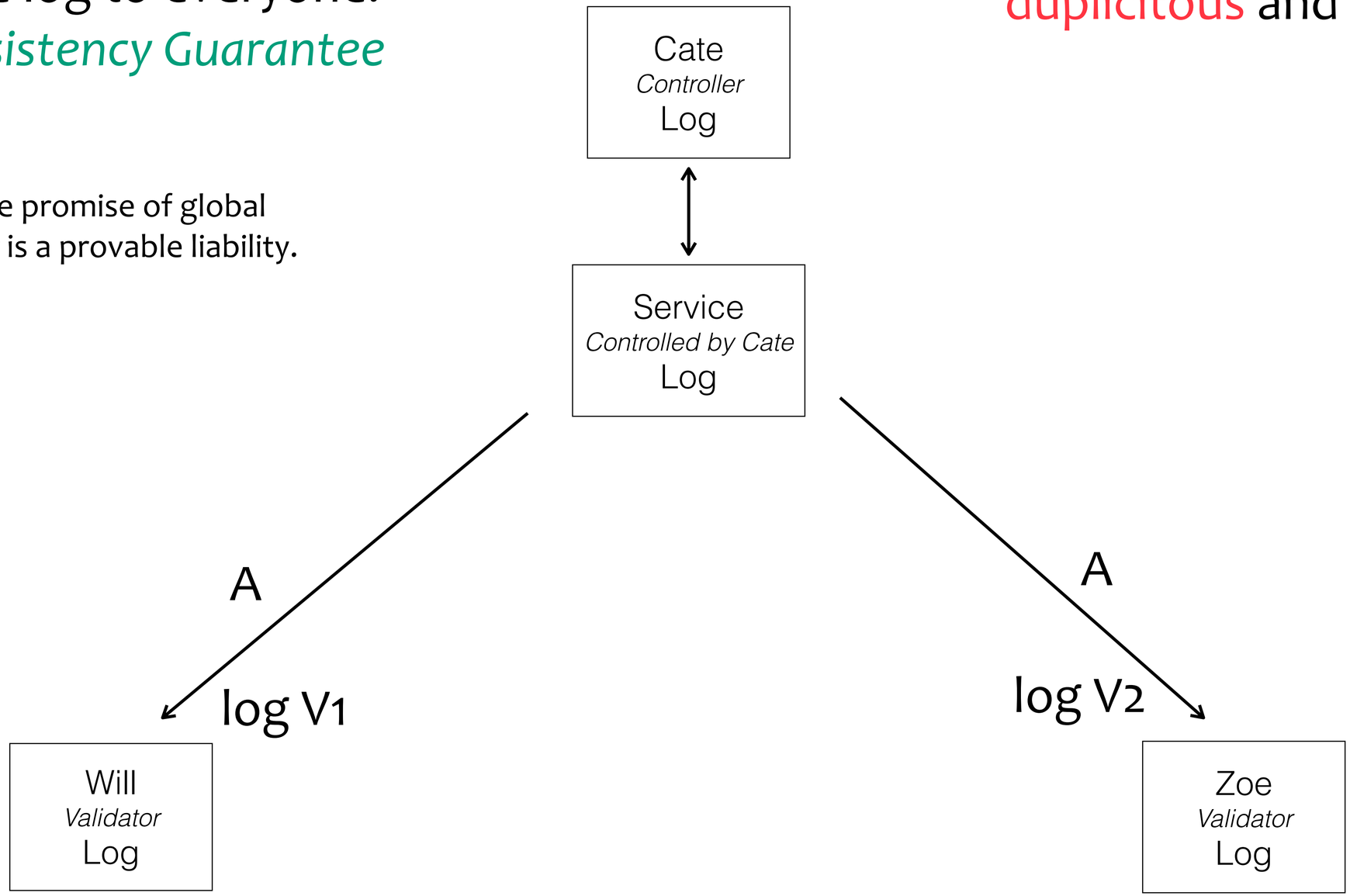
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

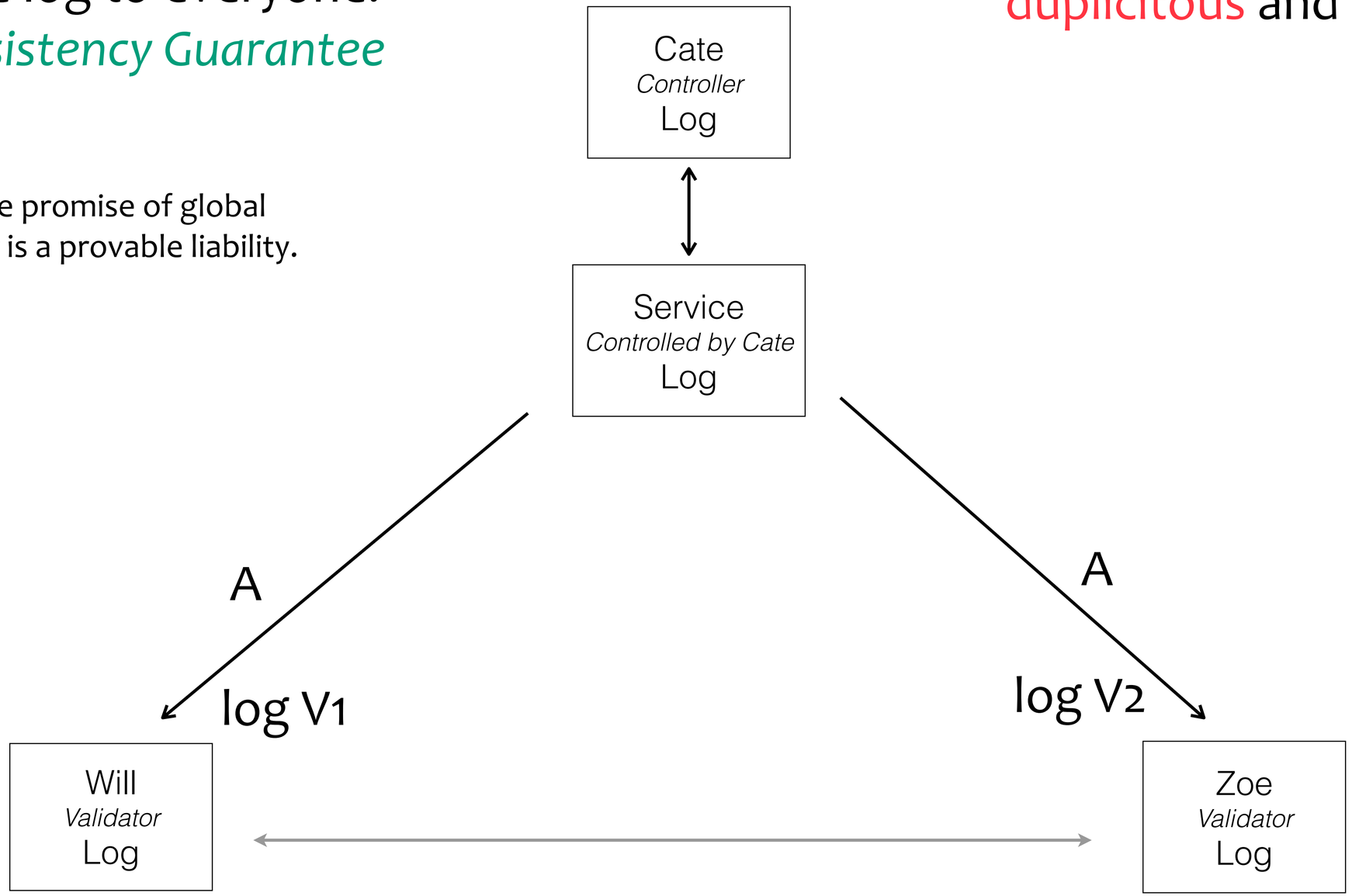
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

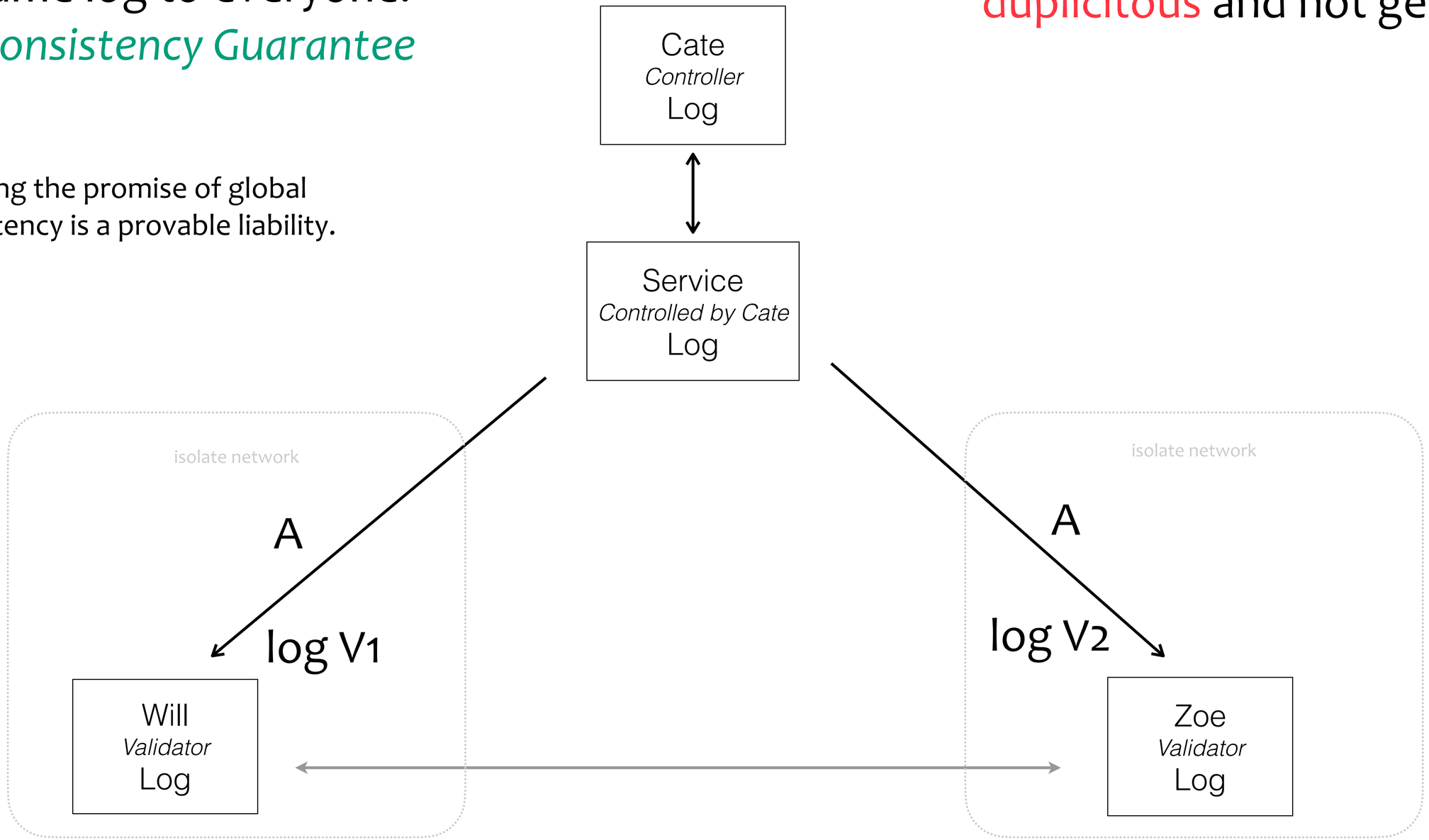
Duplicity Game

Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.



global consistent, highly available, and public (one-to-any) interactions

Duplicity Game

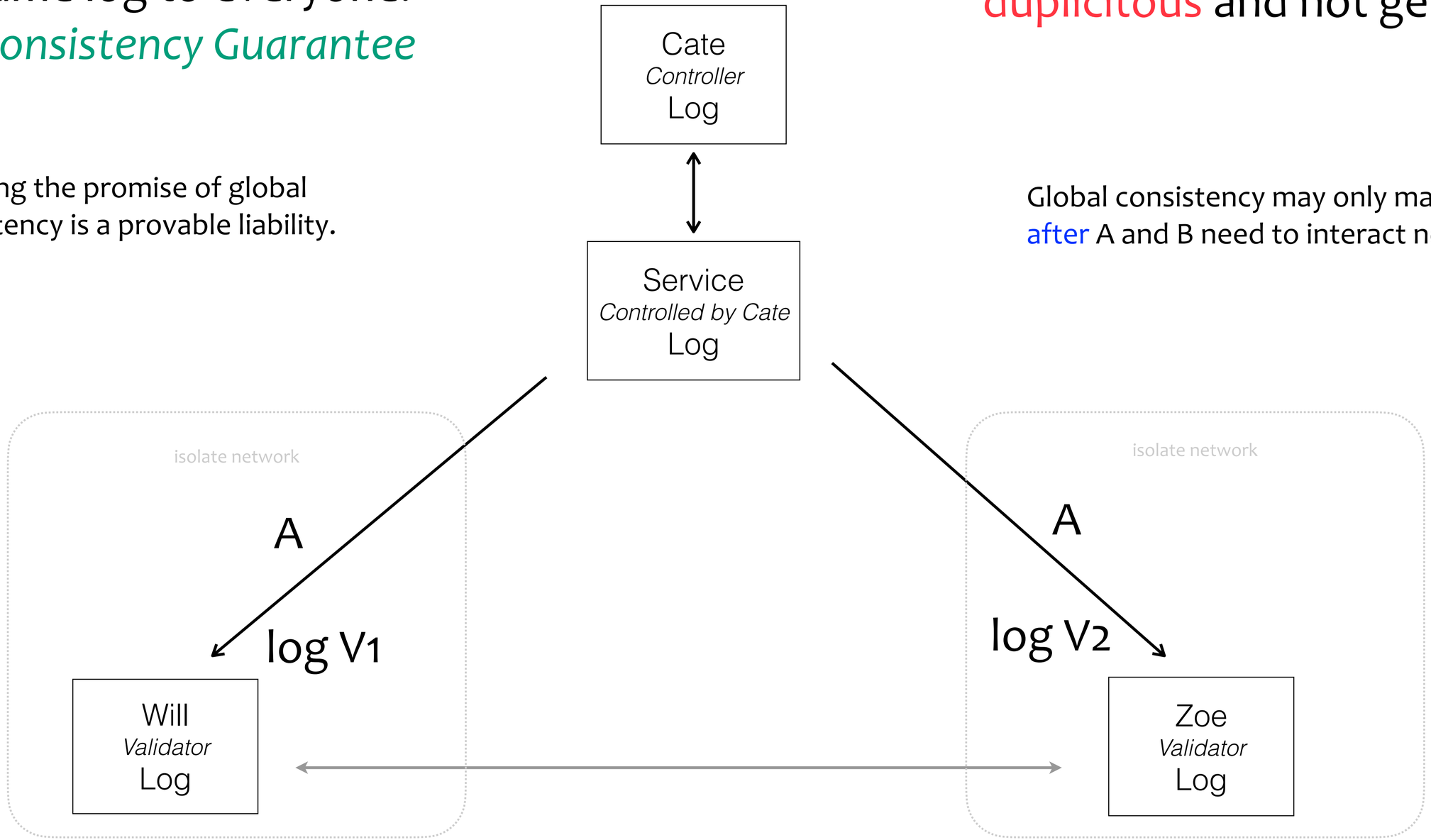
Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.

Global consistency may only matter after A and B need to interact not before.



global consistent, highly available, and public (one-to-any) interactions

Duplicity Game

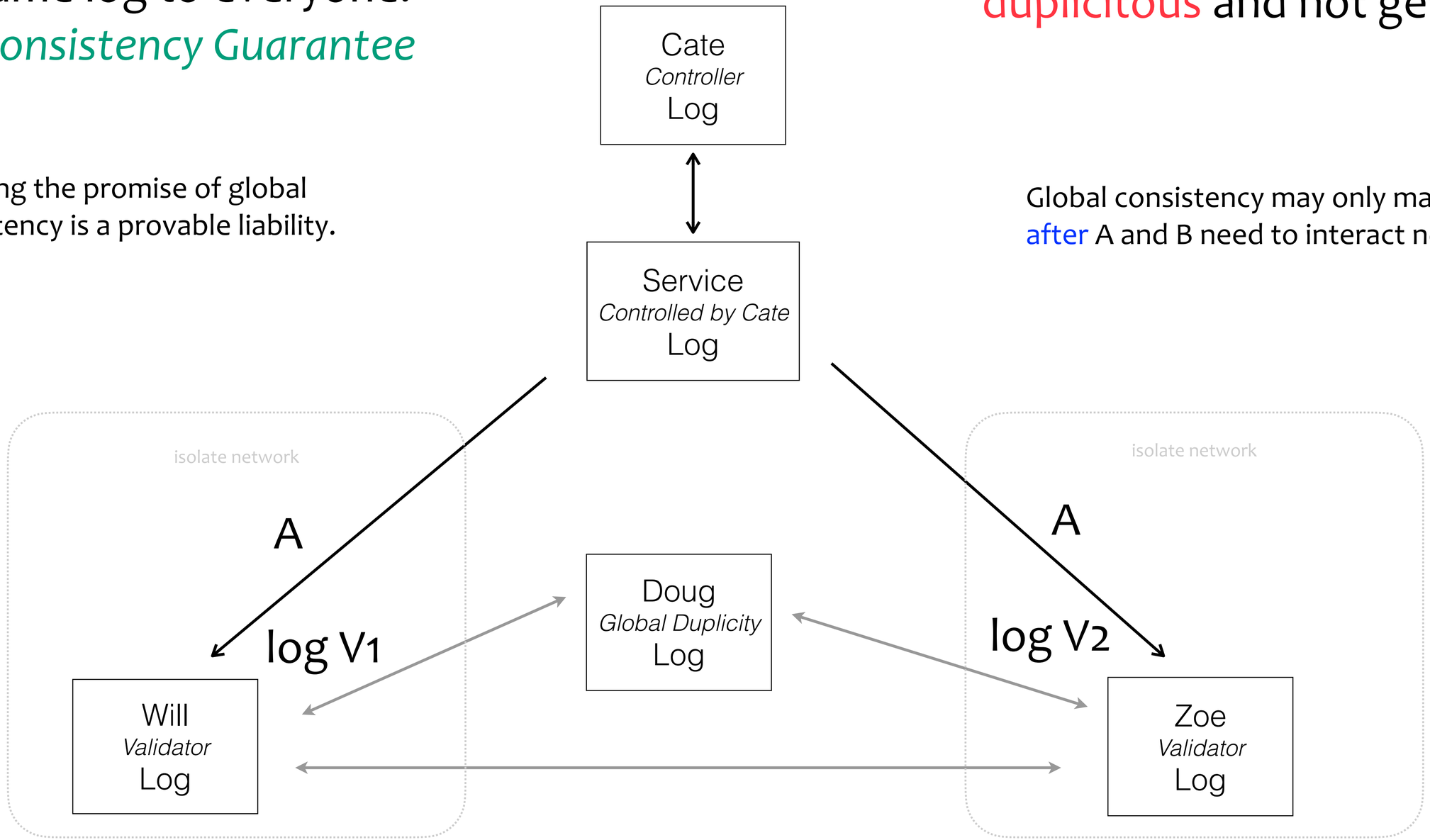
Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.

Global consistency may only matter after A and B need to interact not before.



global consistent, highly available, and public (one-to-any) interactions

Duplicity Game

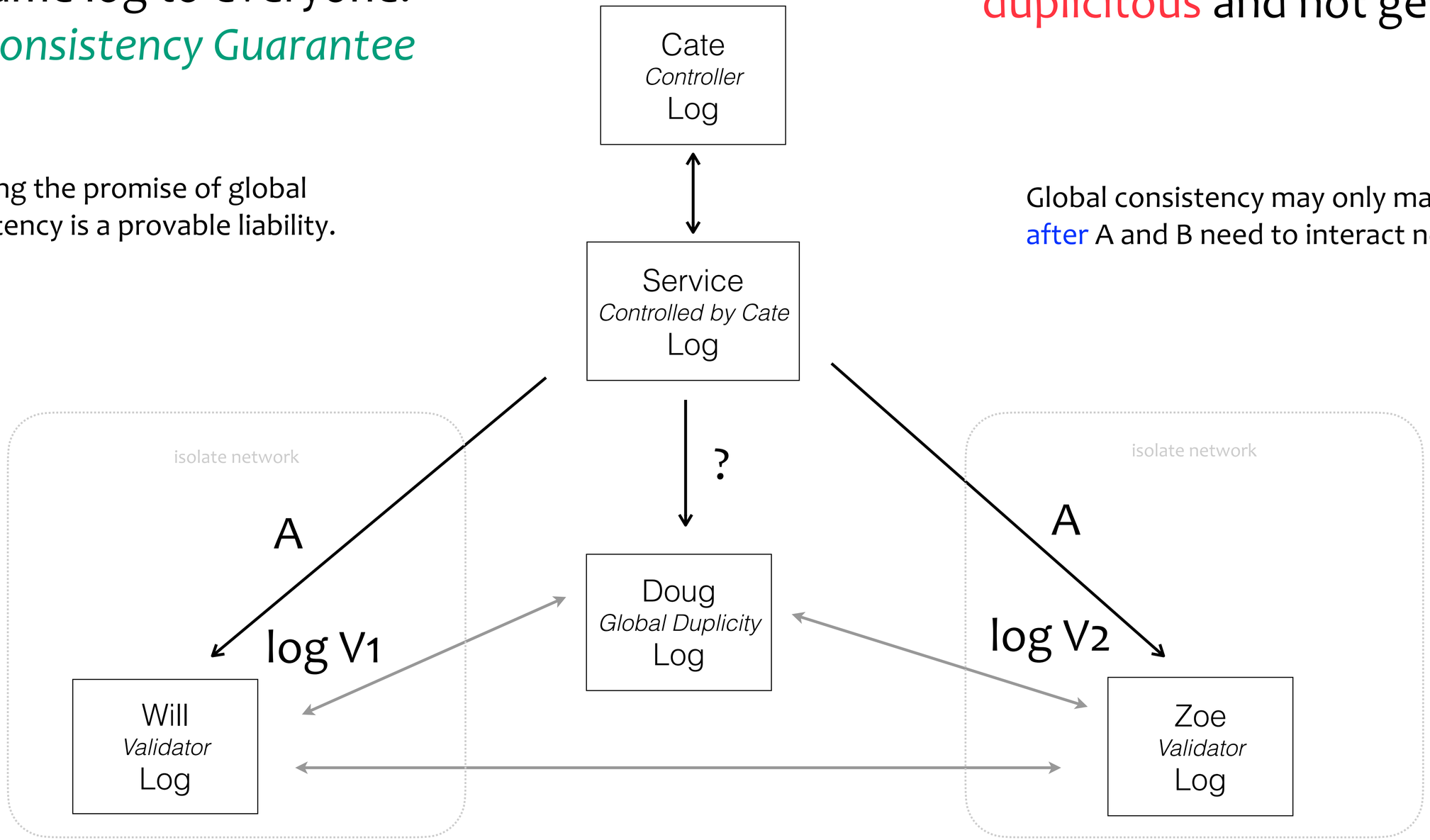
Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.

Global consistency may only matter after A and B need to interact not before.



global consistent, highly available, and public (one-to-any) interactions

Duplicity Game

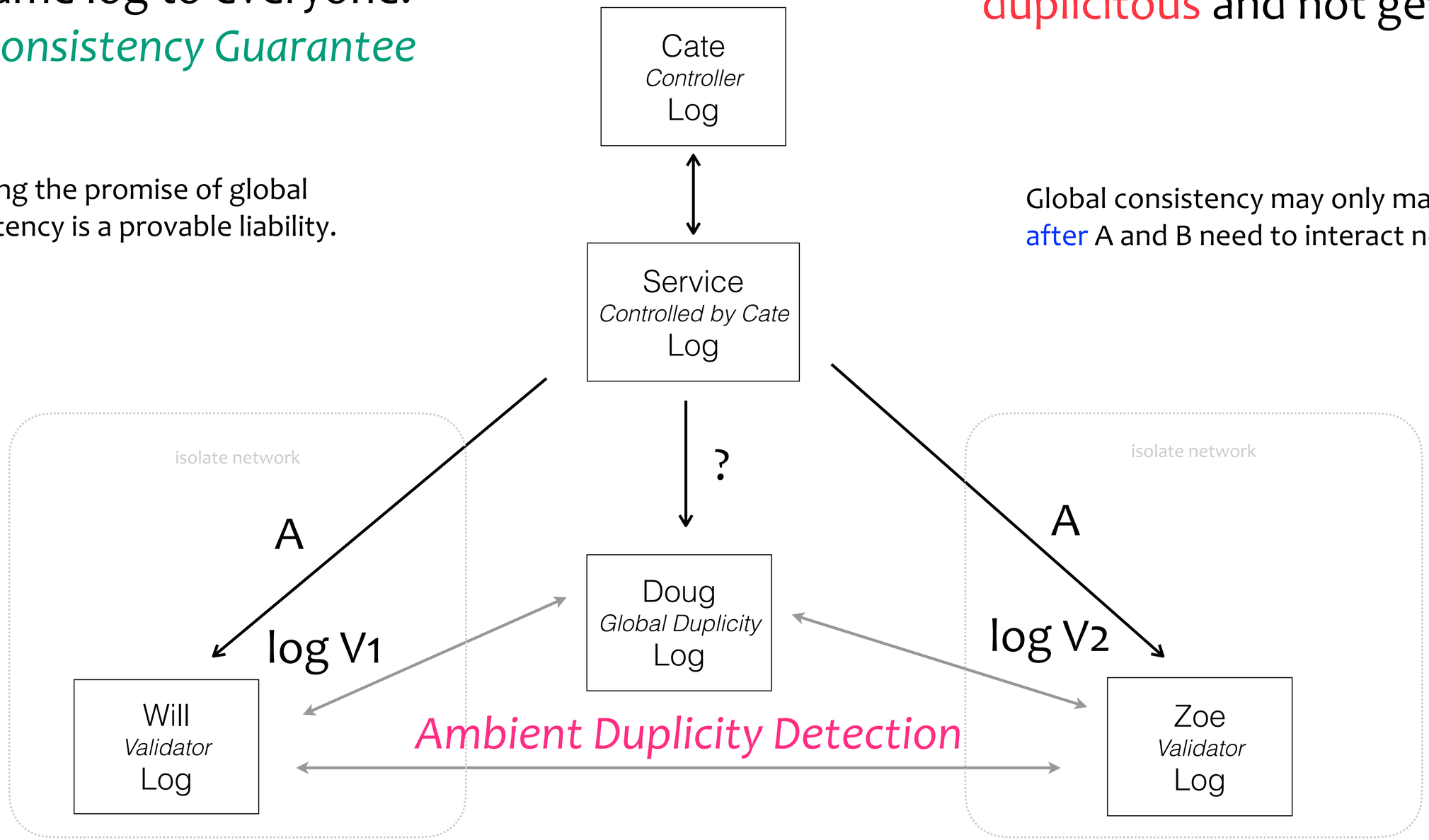
Service promises to provide exact same log to everyone.

Global Consistency Guarantee

How may Cate and/or service be duplicitous and not get caught?

Breaking the promise of global consistency is a provable liability.

Global consistency may only matter after A and B need to interact not before.



global consistent, highly available, and public (one-to-any) interactions

KEY Event Based Provenance of Identifiers

KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority* (*provenance*) for each identifier.

KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority* (*provenance)* for each identifier.

A *proof* is in the form of an identifier's *key event receipt log* (KERL).

KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority* (*provenance*) for each identifier.

A *proof* is in the form of an identifier's *key event receipt log* (KERL).

KERLs are *End Verifiable*:

KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority* (*provenance*) for each identifier.

A *proof* is in the form of an identifier's *key event receipt log* (KERL).

KERLs are *End Verifiable*:

End user alone may verify. Zero trust in intervening infrastructure.

KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority* (*provenance*) for each identifier.

A *proof* is in the form of an identifier's *key event receipt log* (KERL).

KERLs are *End Verifiable*:

End user alone may verify. Zero trust in intervening infrastructure.

KERLs may be *Ambient Verifiable*:

KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority* (*provenance*) for each identifier.

A *proof* is in the form of an identifier's *key event receipt log* (KERL).

KERLs are *End Verifiable*:

End user alone may verify. Zero trust in intervening infrastructure.

KERLs may be *Ambient Verifiable*:

Anyone may verify *anylog*, *anywhere*, at *anytime*.

KEY Event Based Provenance of Identifiers

KERI enables cryptographic *proof-of-control-authority* (*provenance*) for each identifier.

A *proof* is in the form of an identifier's *key event receipt log* (KERL).

KERLs are *End Verifiable*:

End user alone may verify. Zero trust in intervening infrastructure.

KERLs may be *Ambient Verifiable*:

Anyone may verify *anylog*, *anywhere*, at *anytime*.

KERI = self-cert root-of-trust + certificate transparency + KA^{2}CE + recoverable + post-quantum.

BACKGROUND

Certificate Transparency Problem

“The solution the computer world has relied on for many years is to introduce into the system trusted third parties (CAs) that vouch for the binding between the domain name and the private key. The problem is that we've managed to bless several hundred of these supposedly trusted parties, any of which can vouch for any domain name. Every now and then, one of them gets it wrong, sometimes spectacularly.”

Pinning inadequate
Notaries inadequate
DNSSec inadequate
All require trust in 3rd party compute infrastructure that is inherently vulnerable

Certificate Transparency: (related EFF SSL Observatory)
Public end-verifiable append-only event log with consistency and inclusion proofs
End-verifiable duplicity detection = Ambient verifiability of duplicity
Event log is third party infrastructure but zero trust because it is verifiable.
Sparse Merkle Trees for revocation of certificates

Certificate Transparency Solution

Public end-verifiable append-only event log with consistency and inclusion proofs
End-verifiable duplicity detection = ambient verifiability of duplicity
Event log is third party infrastructure but it is not trusted because logs are verifiable.
Sparse Merkle trees for revocation of certificates
(related EFF SSL Observatory)

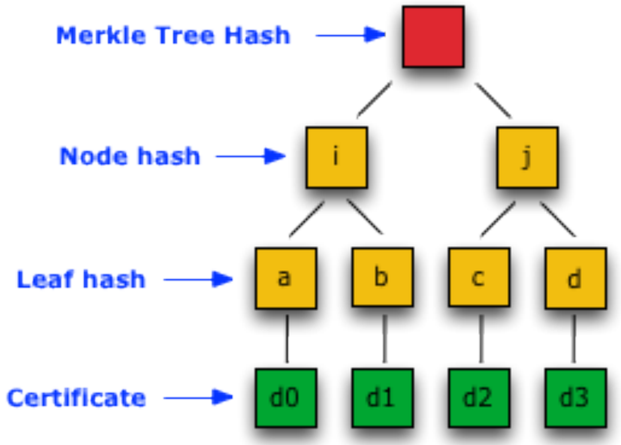


Figure 1

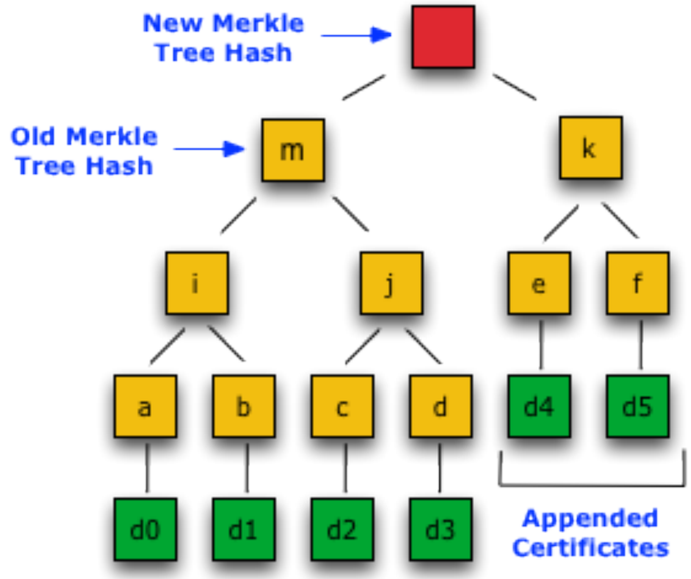


Figure 2

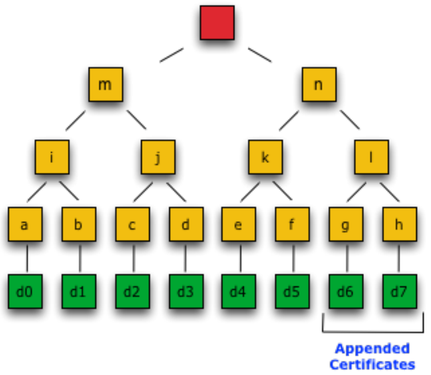


Figure 3

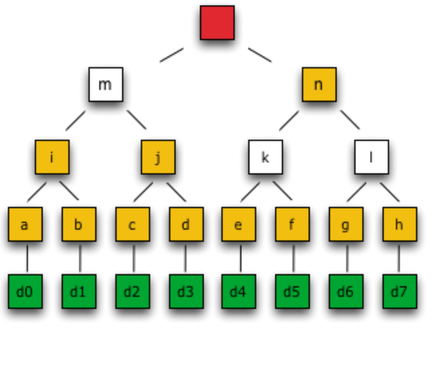


Figure 4