Transparency Overlays and Applications

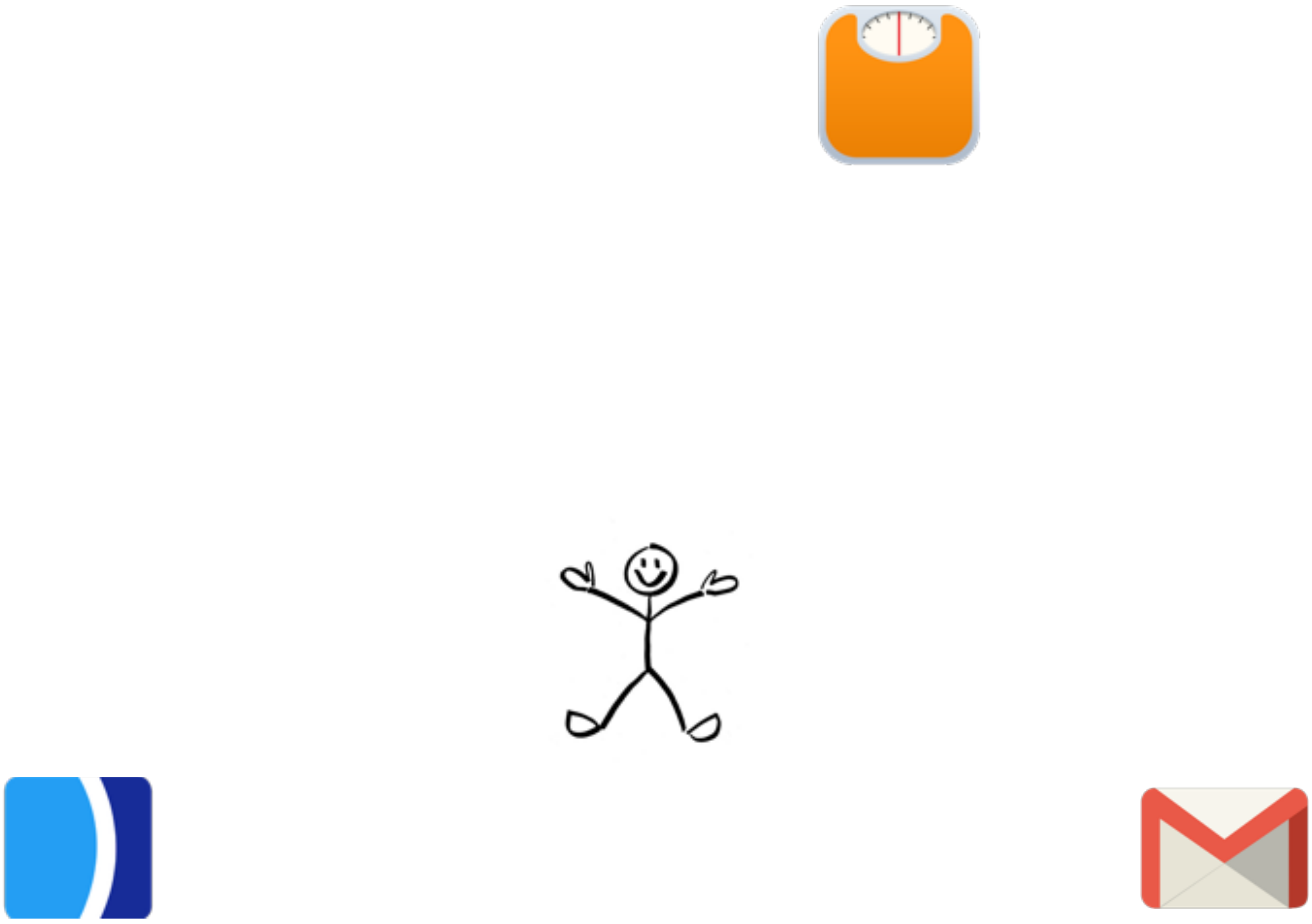
Melissa Chase (Microsoft Research Redmond)

Sarah Meiklejohn (University College London)

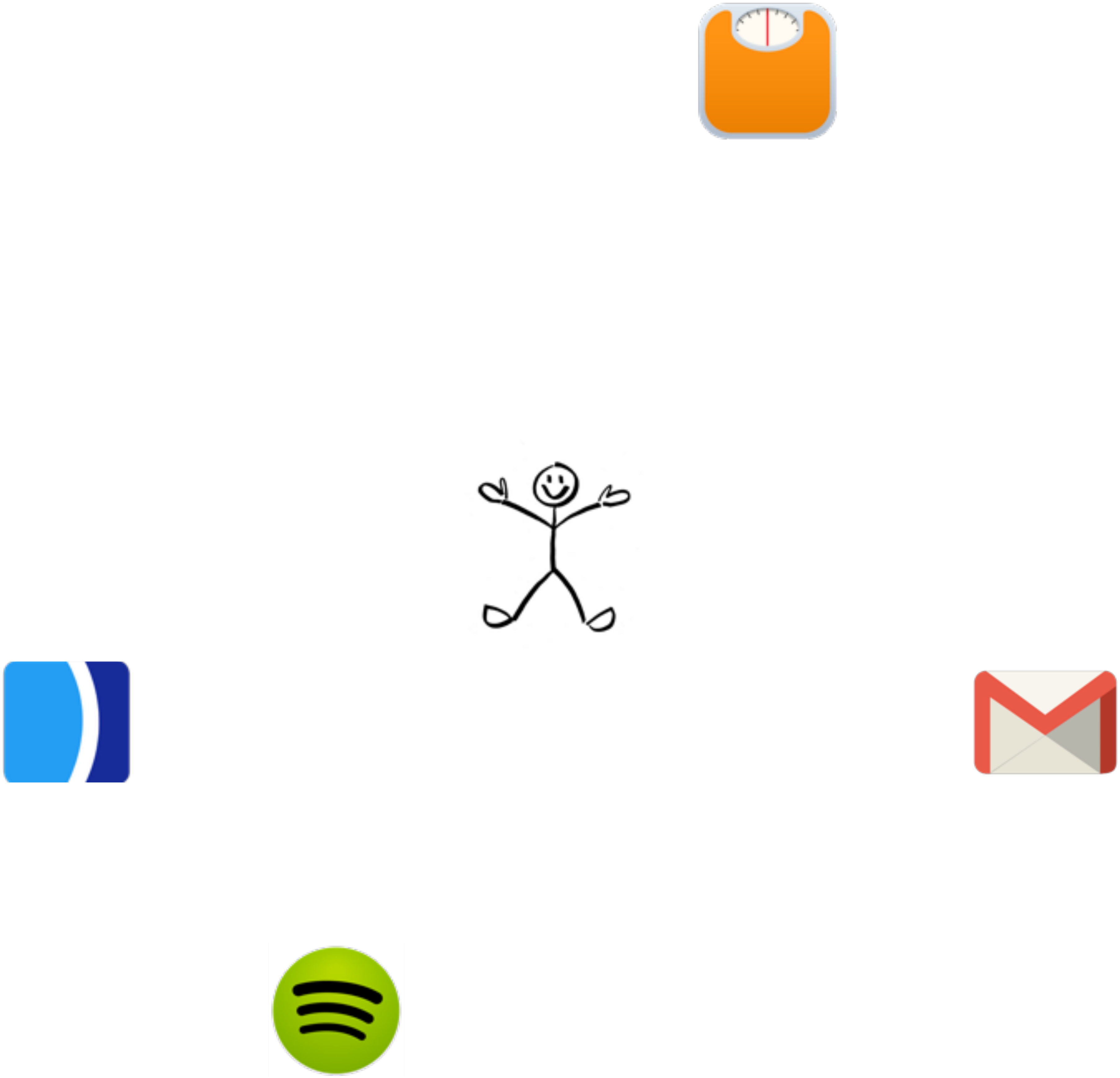
(icons by parkjisun from noun project)

(icons by parkjisun from noun project)

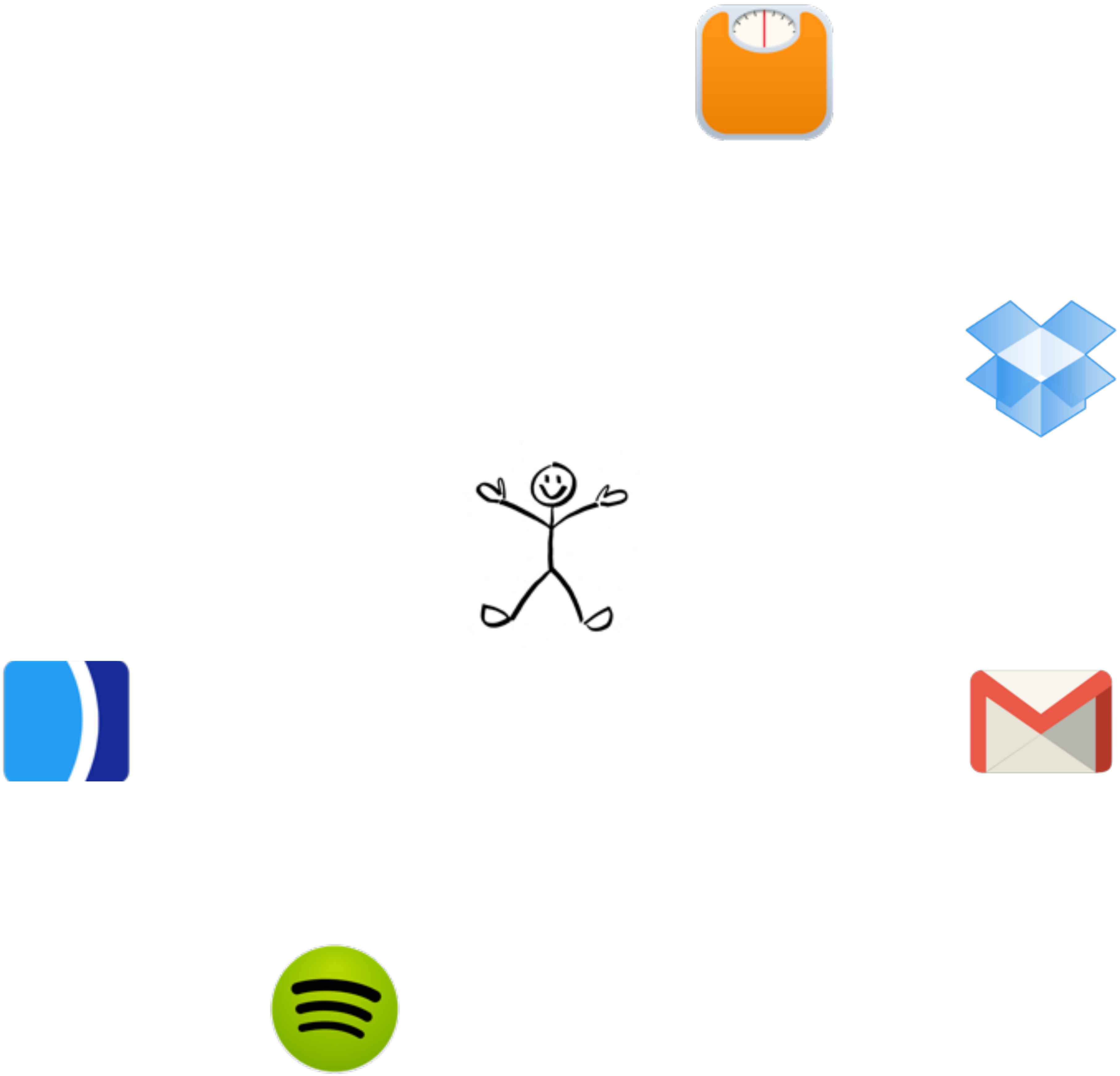
(icons by parkjisun from noun project)



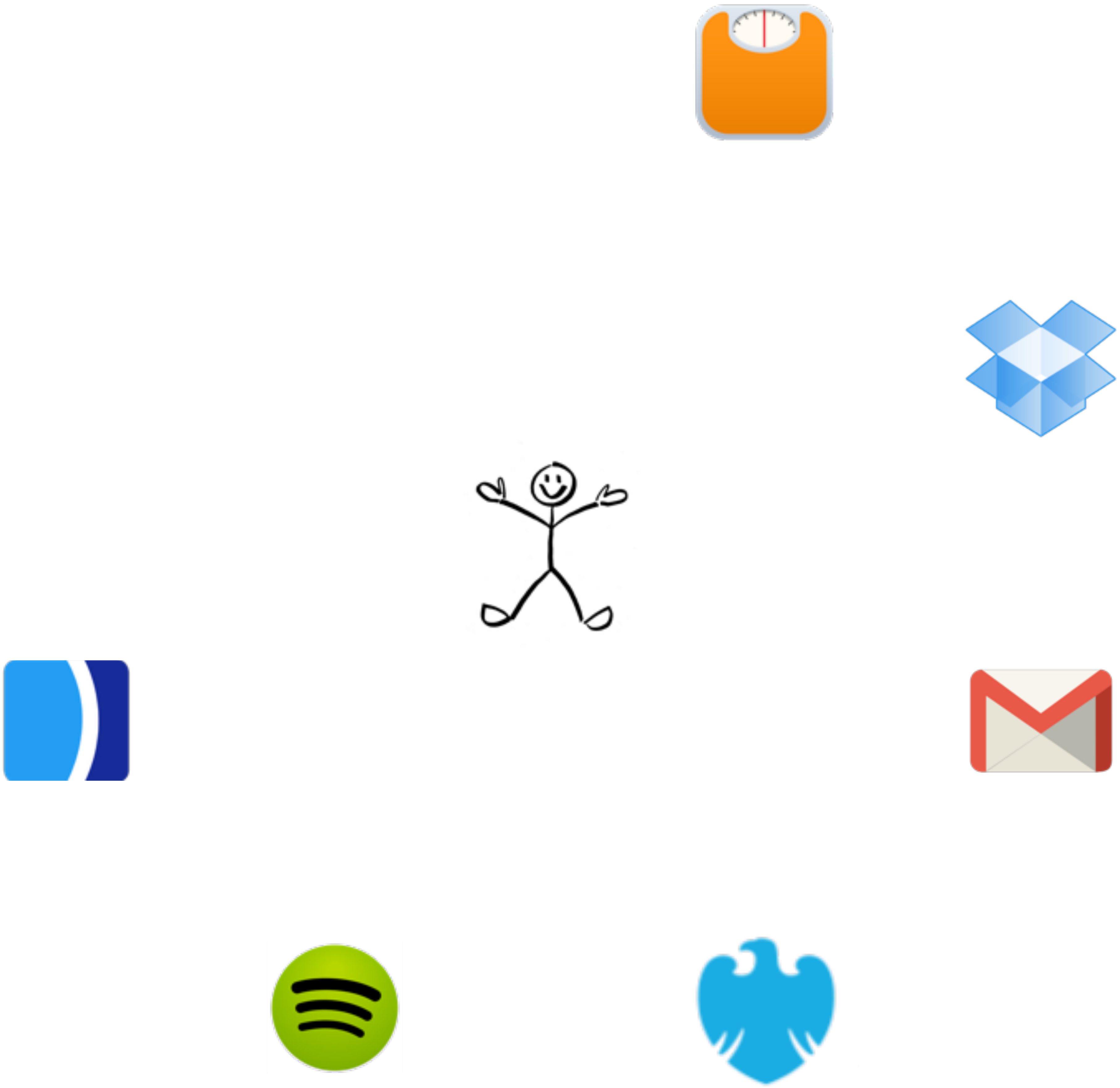
(icons by parkjisun from noun project)



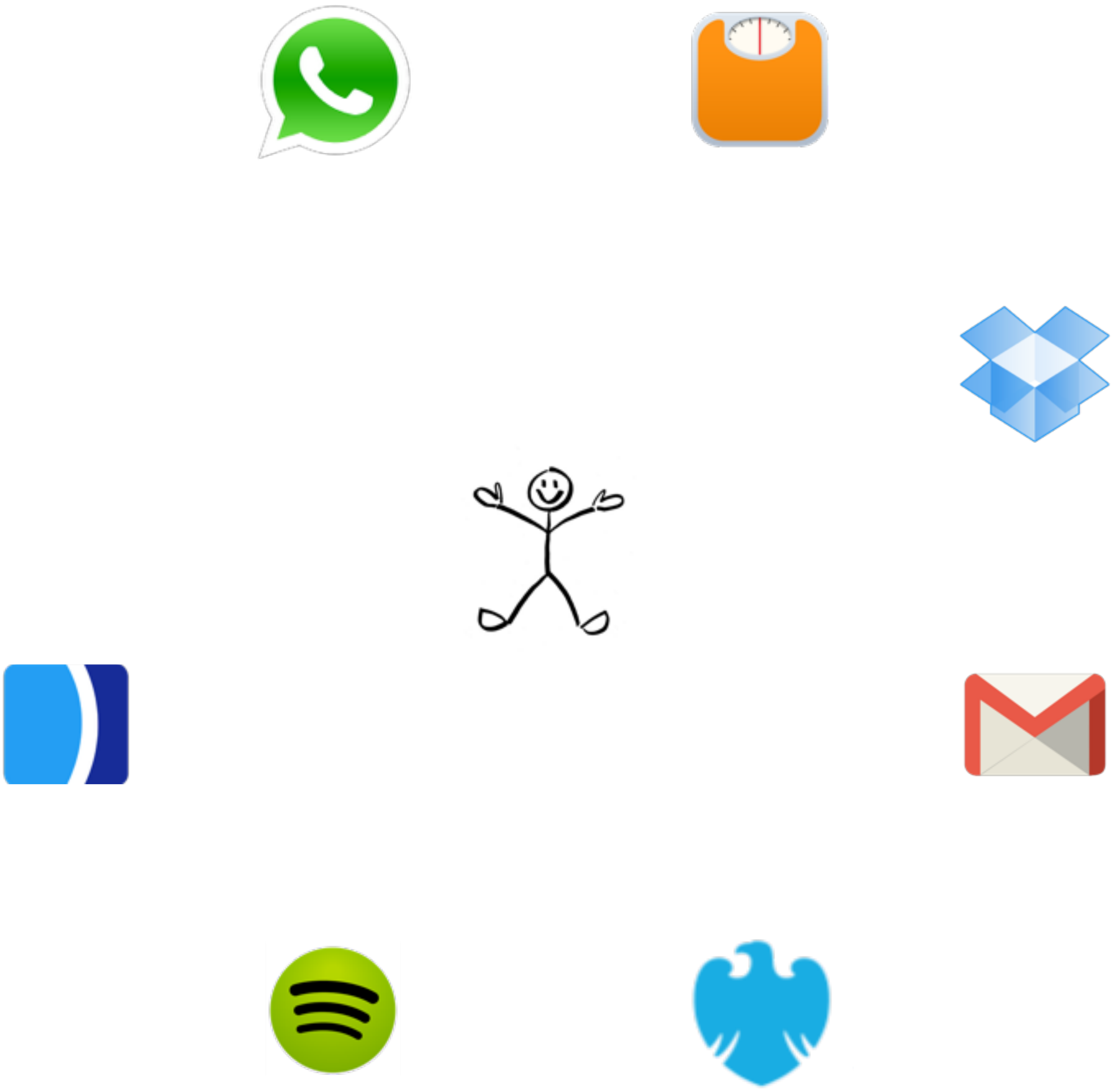
(icons by parkjisun from noun project)



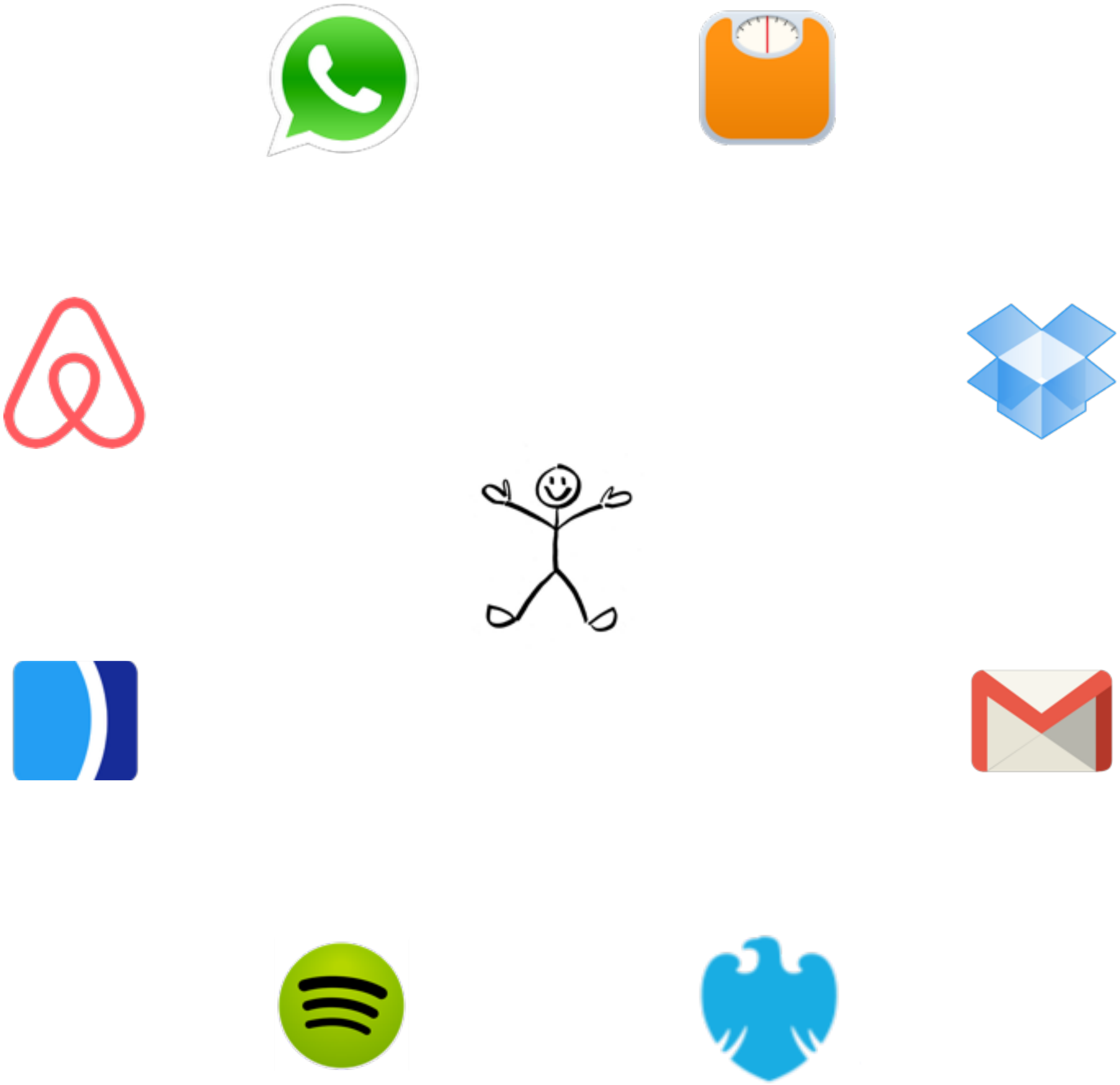
(icons by parkjisun from noun project)



(icons by parkjisun from noun project)

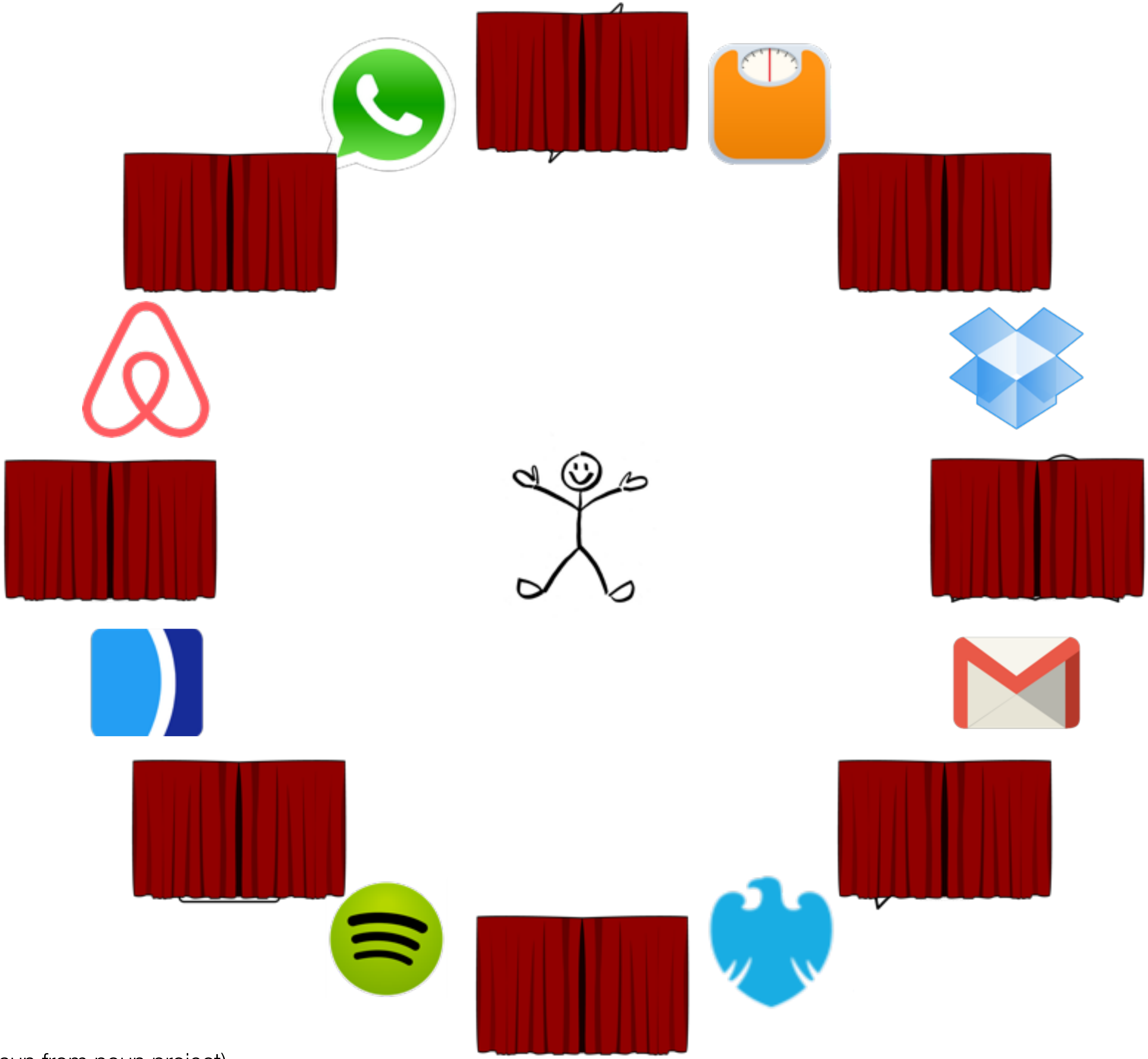


(icons by parkjisun from noun project)

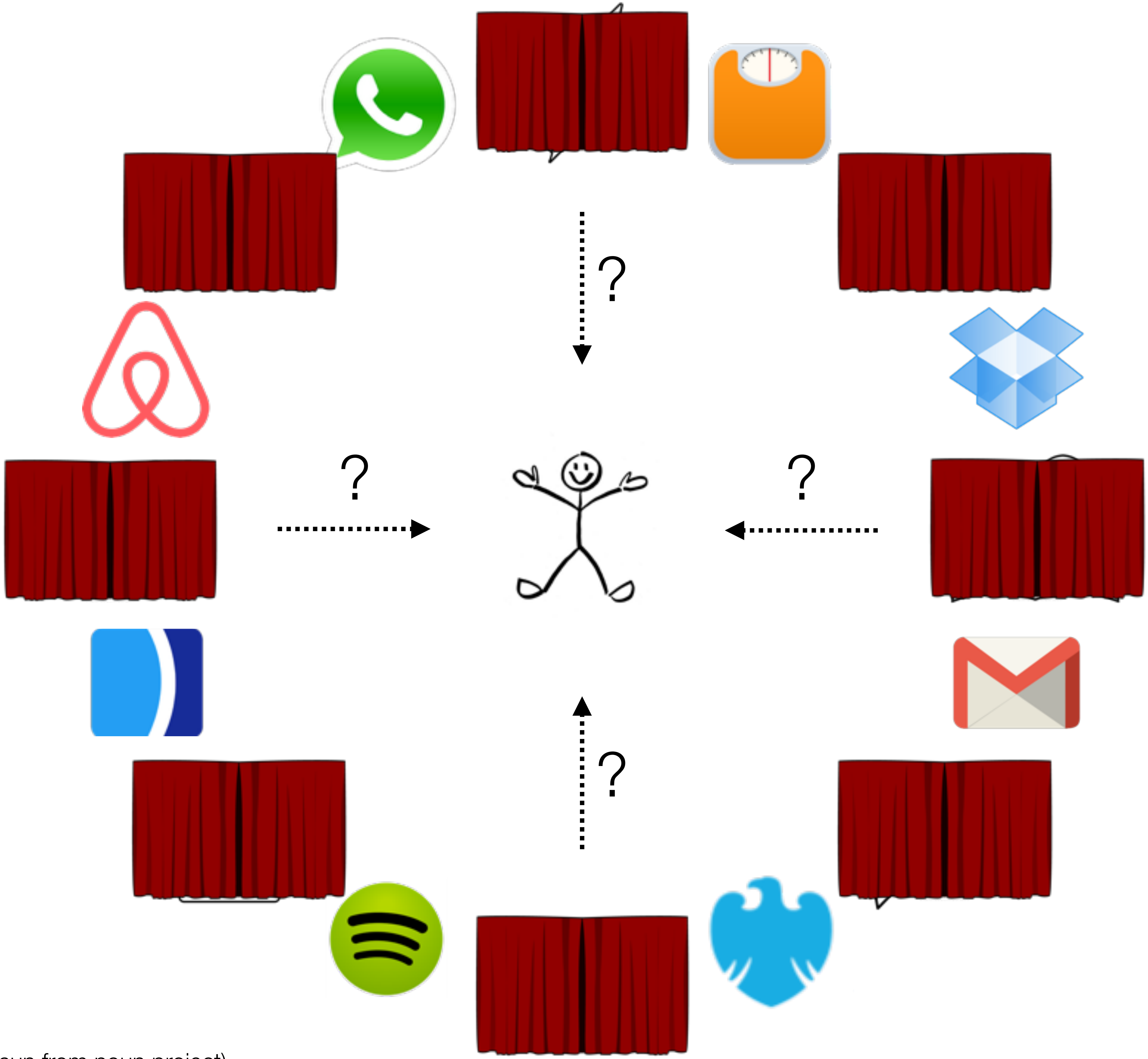


(icons by parkjisun from noun project)

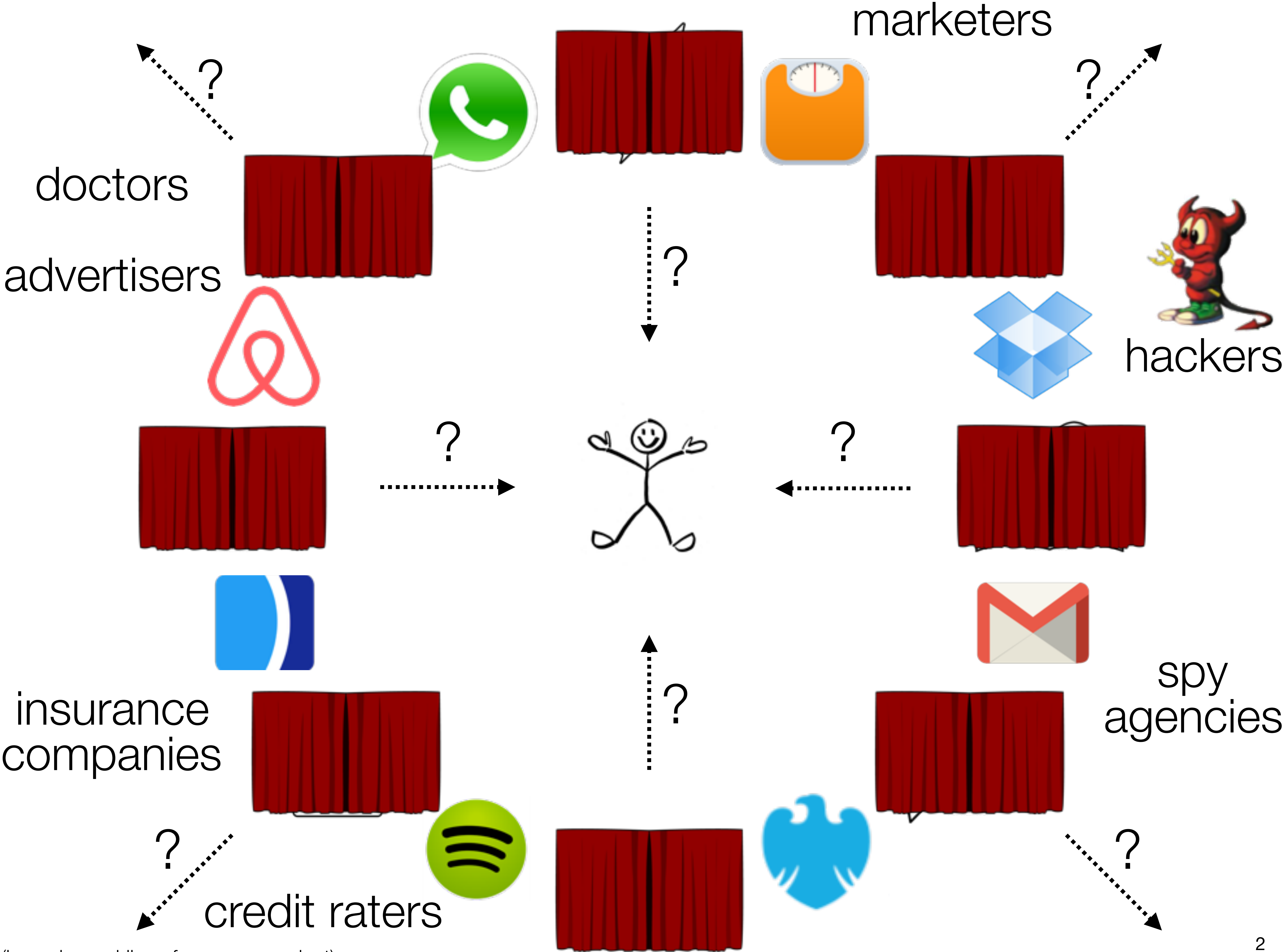
(icons by parkjisun from noun project)



(icons by parkjisun from noun project)



(icons by parkjisun from noun project)



marketers

doctors

advertisers

hackers

spy agencies

insurance companies

credit raters

?

?

?

?

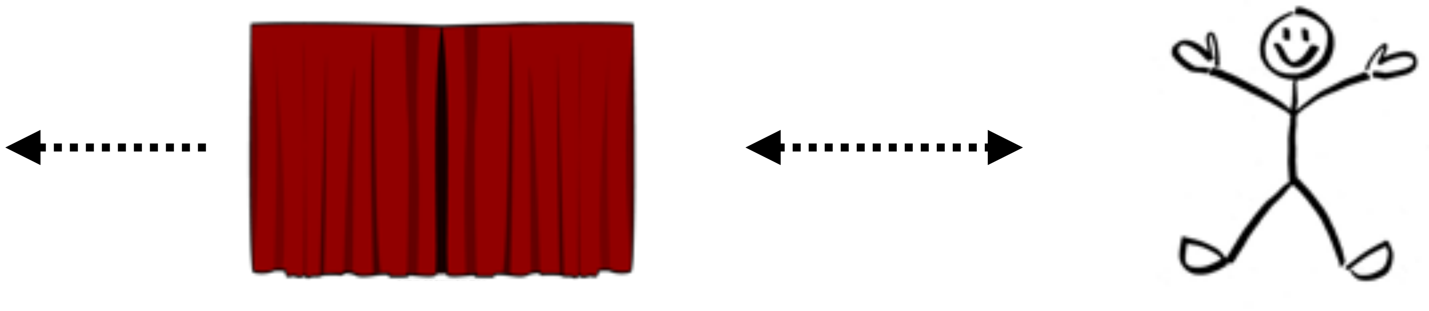
?

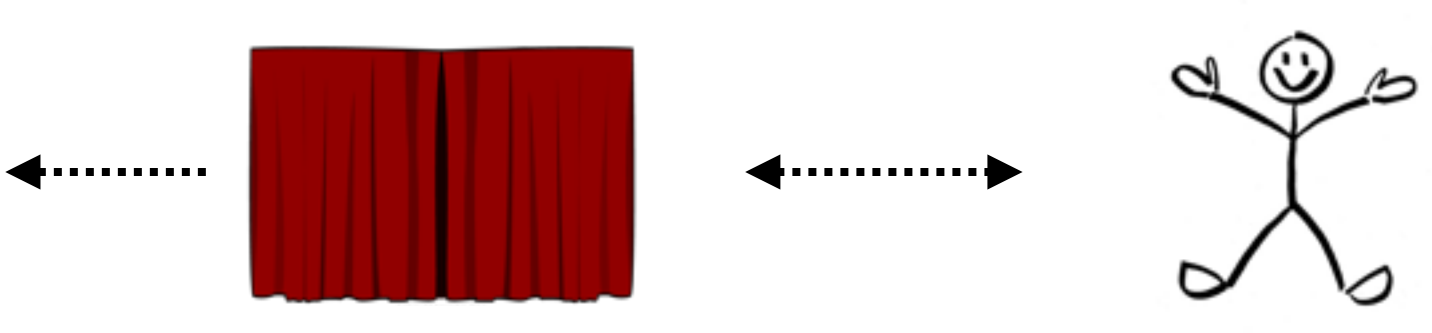
?

?

?

(icons by parkjisun from noun project)



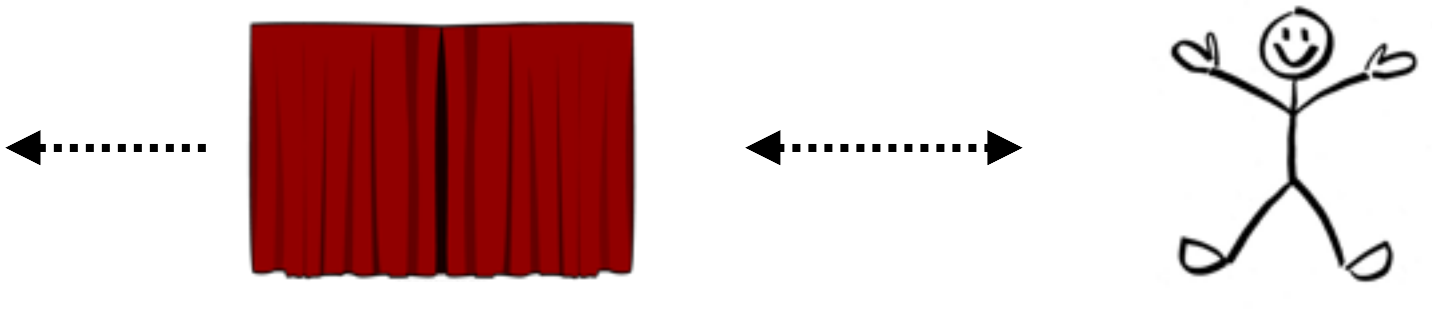


events in the system can be

-data access by user

-data access by third party

-data creation by user

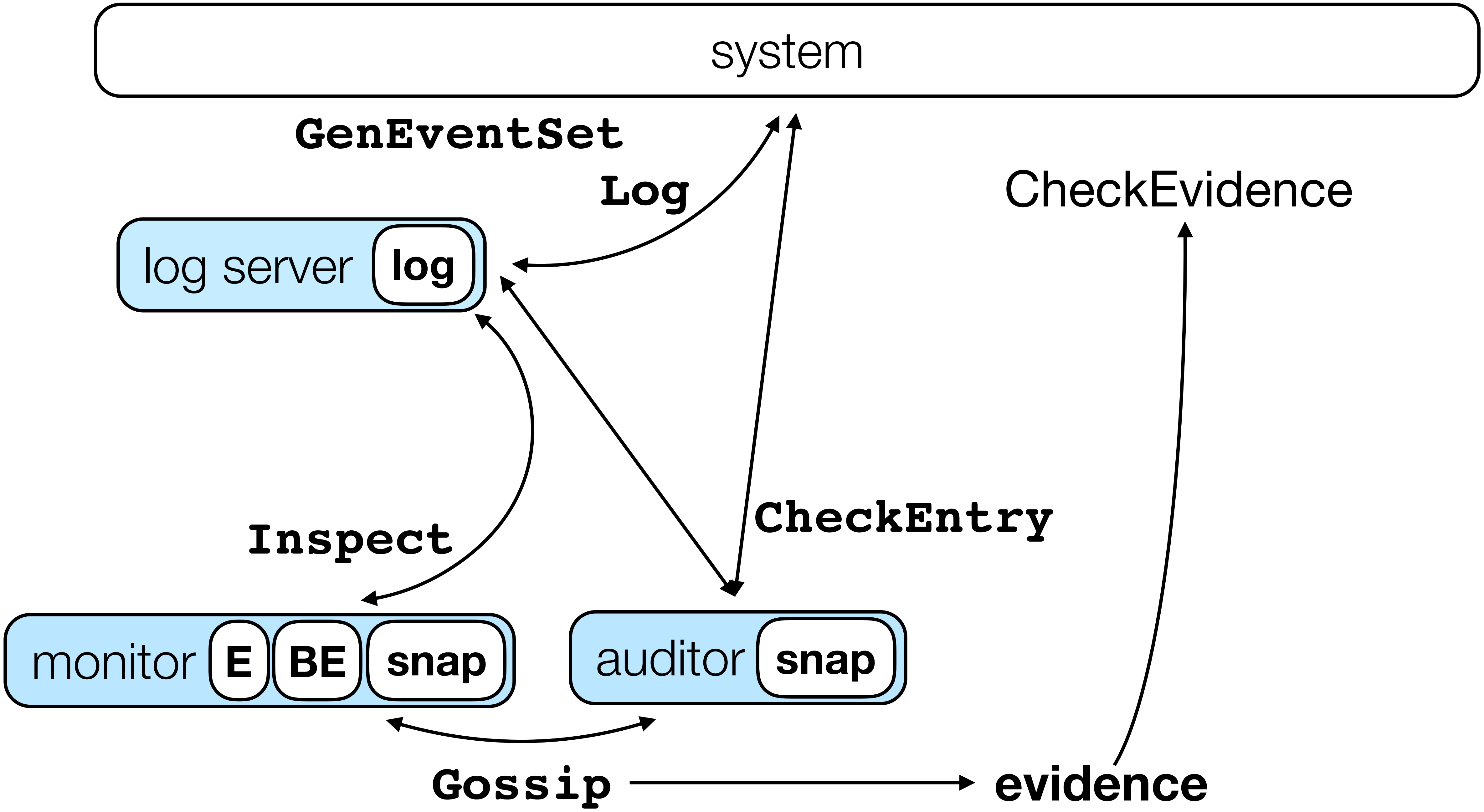


events in the system can be

- -data access by user
- -data access by third party
- -data creation by user

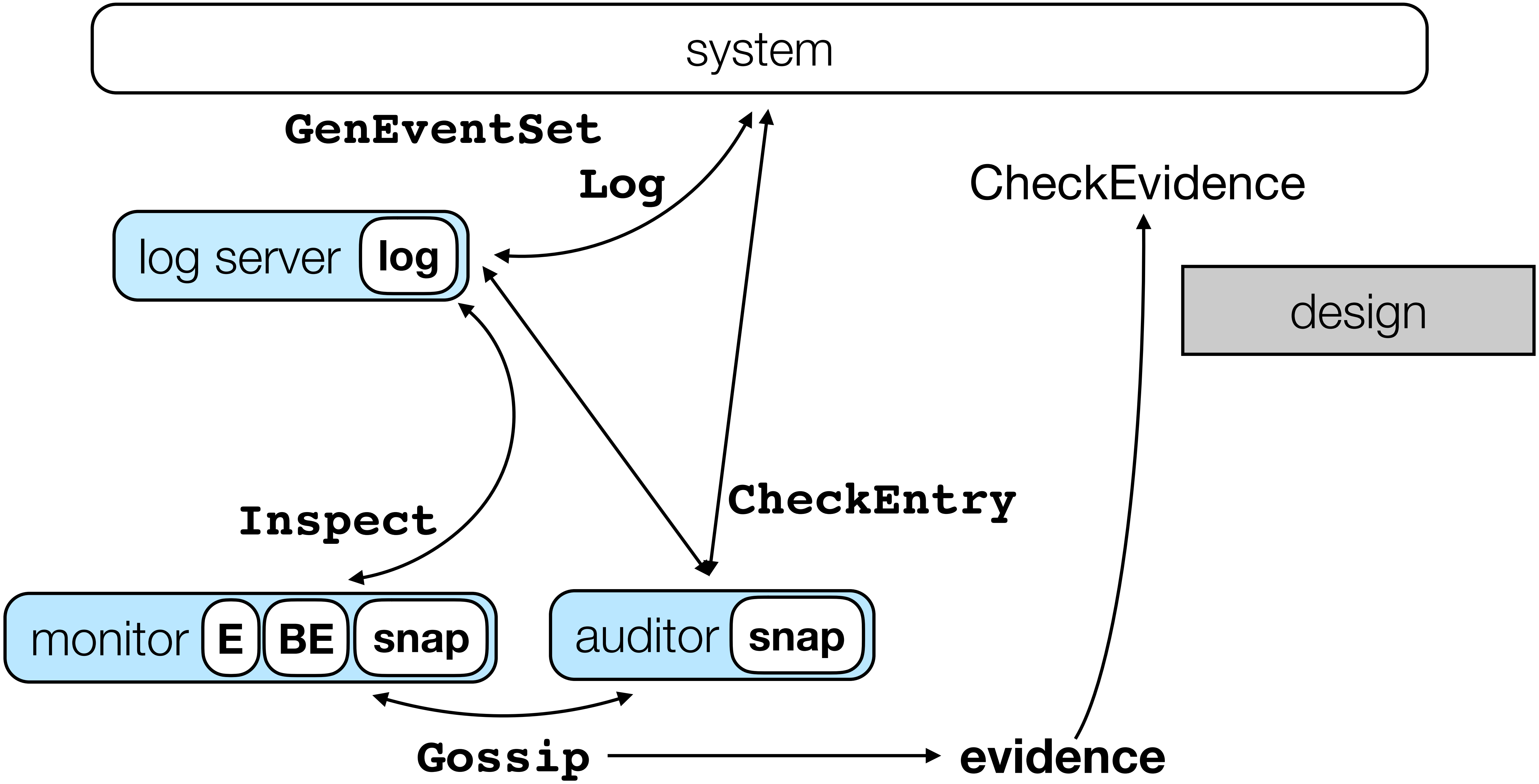
transparency: bad events are exposed

a transparency overlay



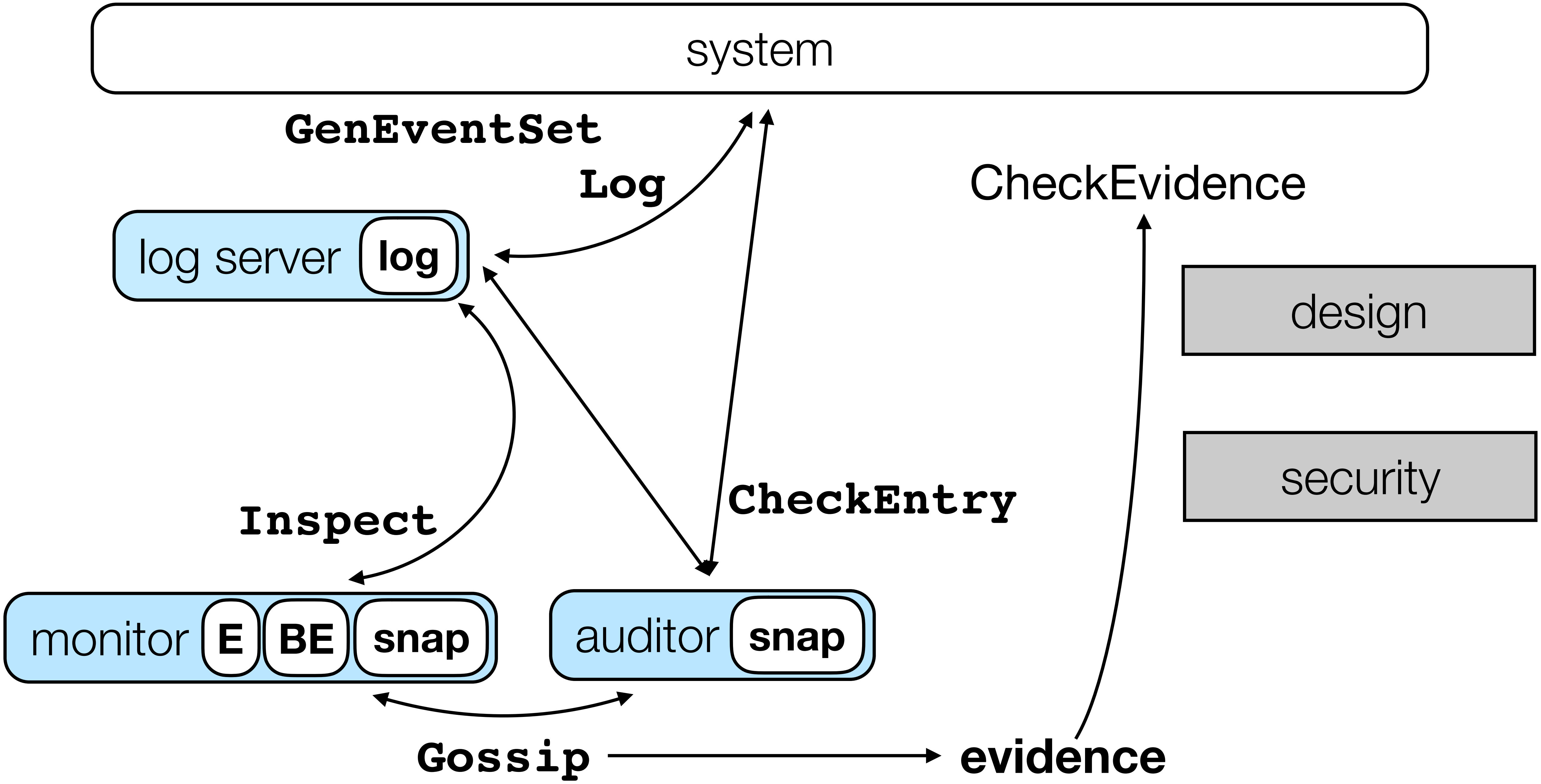
(architecture very much inspired by Certificate Transparency [LL'13])

a transparency overlay



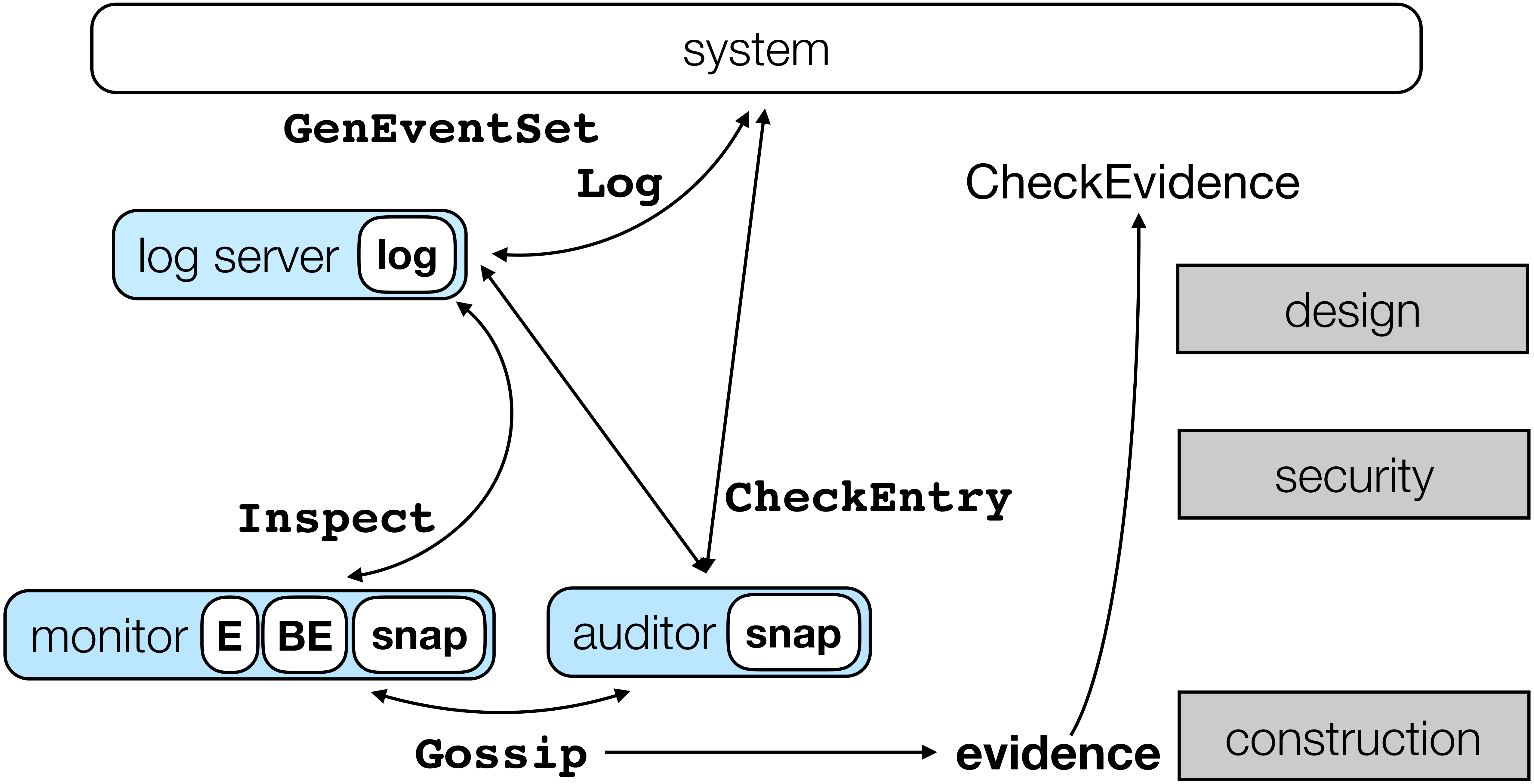
(architecture very much inspired by Certificate Transparency [LL'13])

a transparency overlay



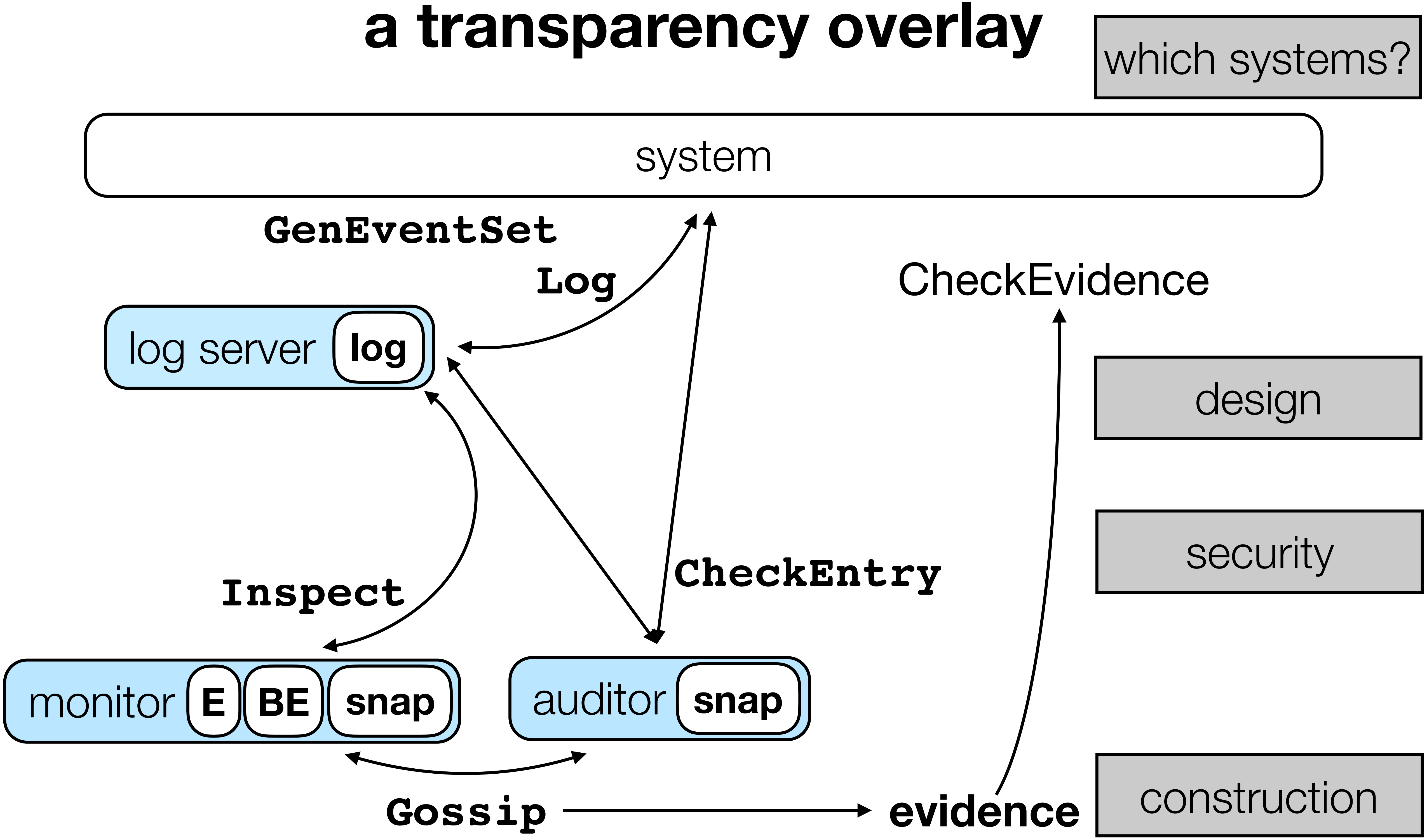
(architecture very much inspired by Certificate Transparency [LL'13])

a transparency overlay



(architecture very much inspired by Certificate Transparency [LL'13])

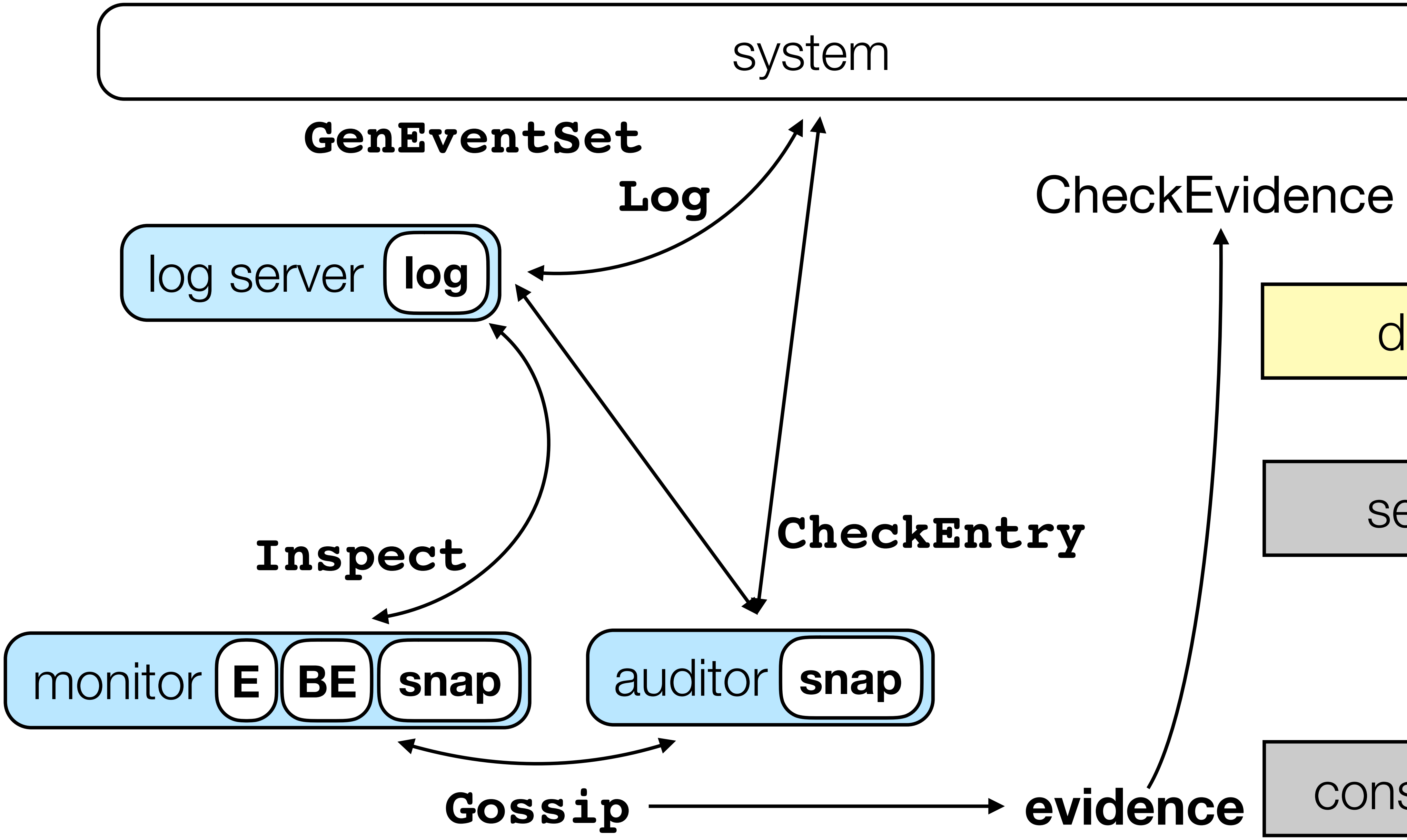
a transparency overlay



(architecture very much inspired by Certificate Transparency [LL'13])

a transparency overlay

which systems?



system

GenEventSet

Log

CheckEvidence

log server

log

Inspect

CheckEntry

monitor

E

BE

snap

auditor

snap

Gossip

evidence

design

security

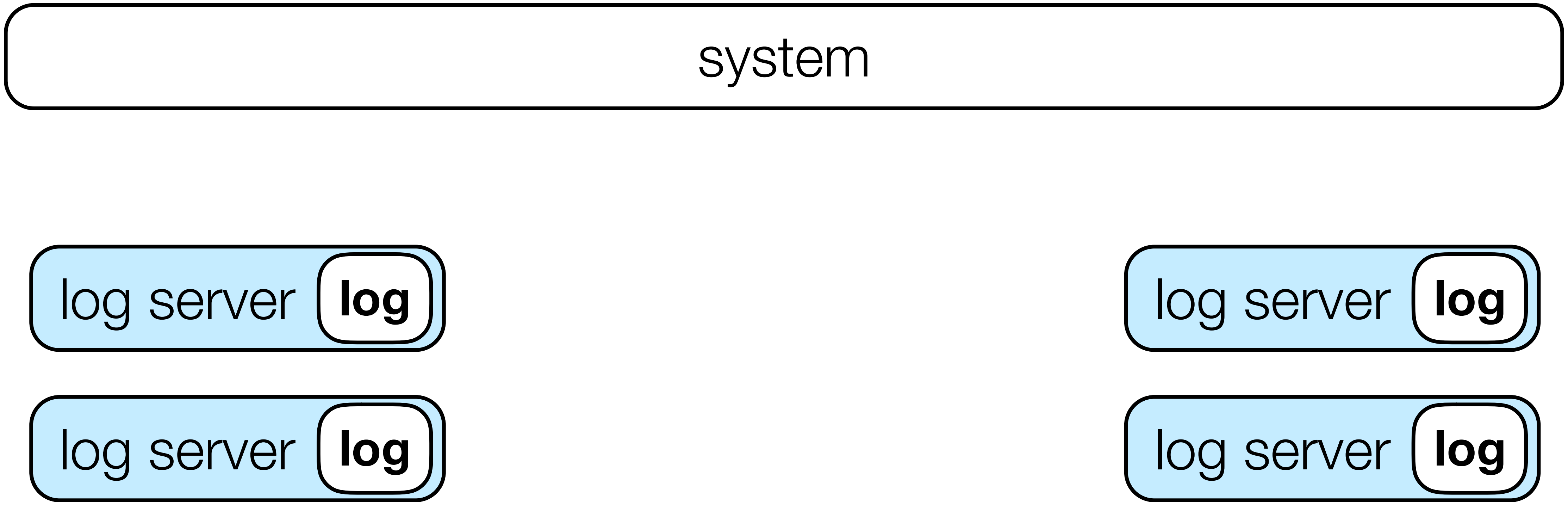
construction

(architecture very much inspired by Certificate Transparency [LL'13])

design

system

design



system

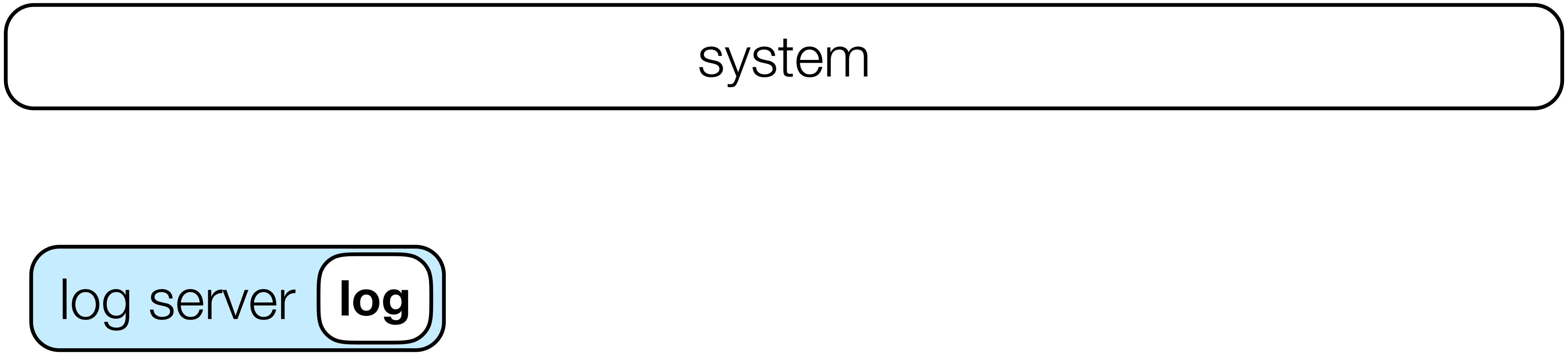
log server log

log server log

log server log

log server log

design

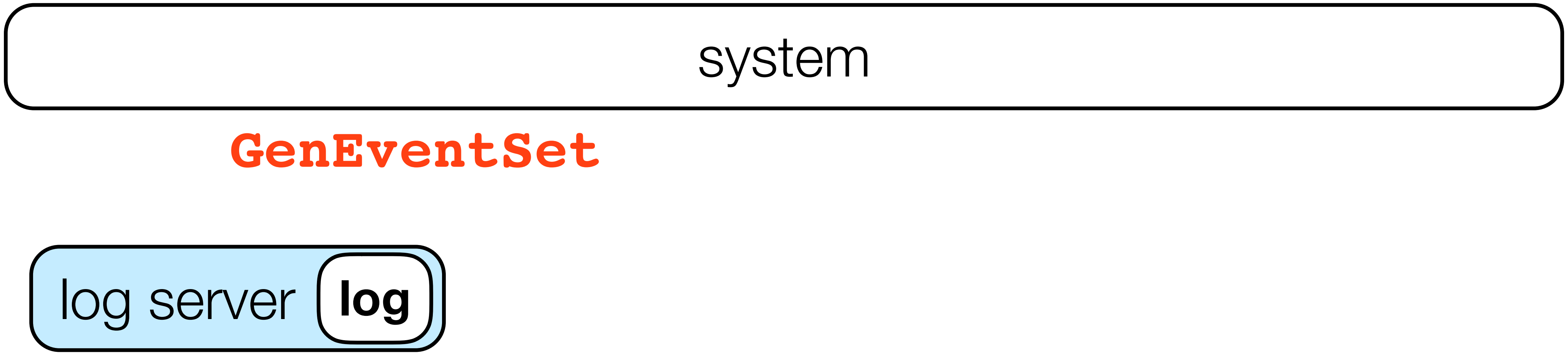


system

log server

log

design



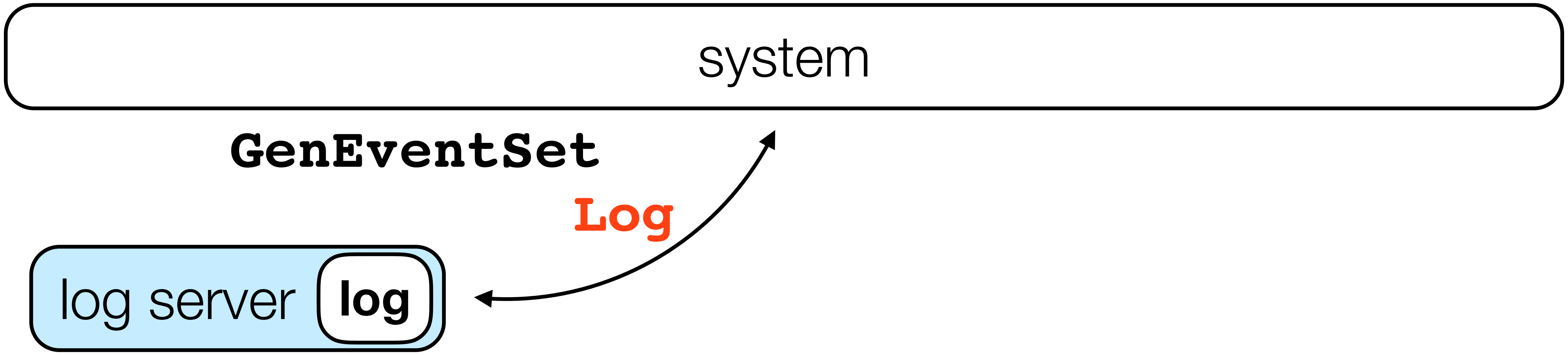
system

GenEventSet

log server

log

design



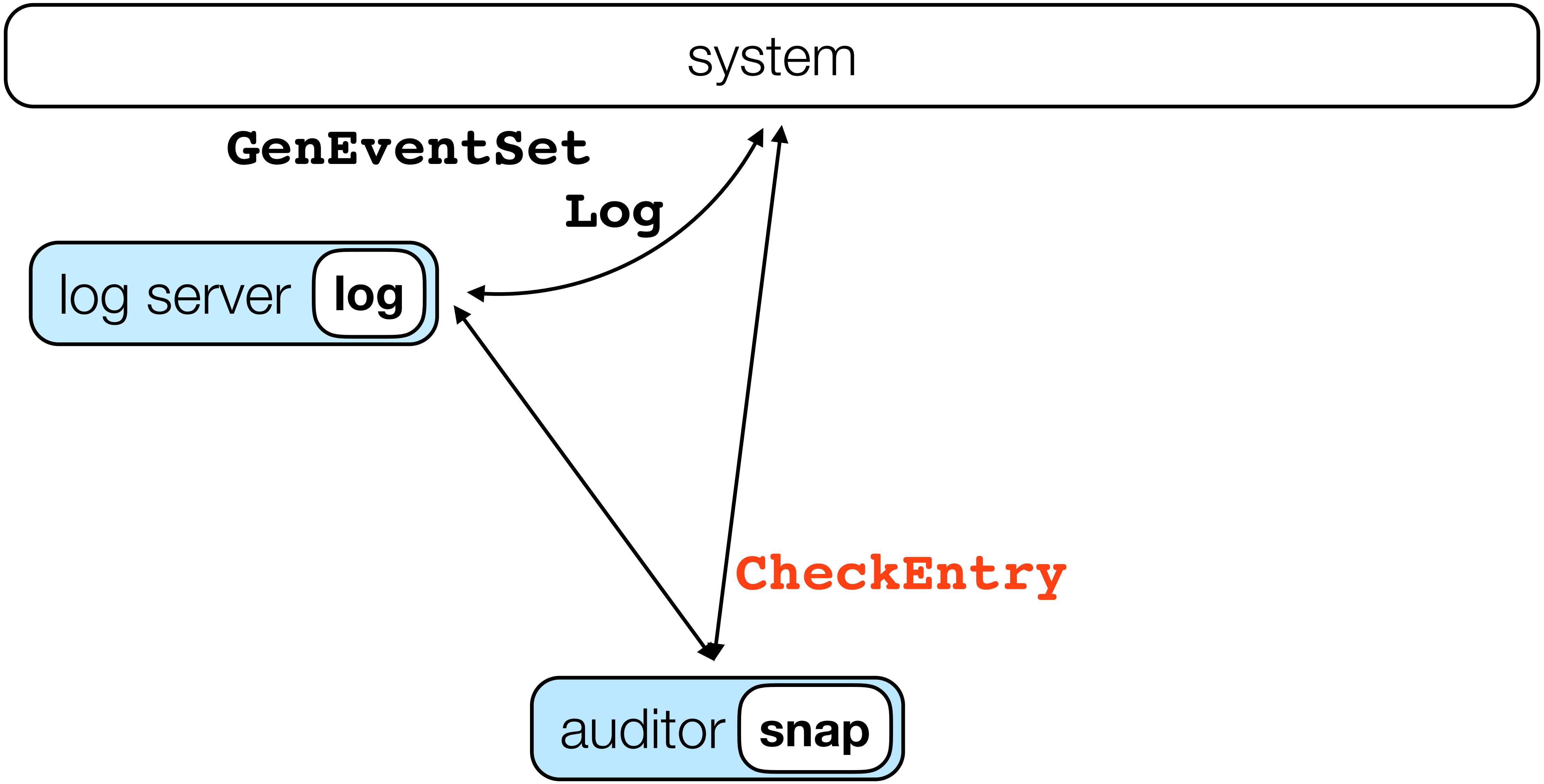
system

GenEventSet

Log

log server

log



system

`GenEventSet`

`Log`

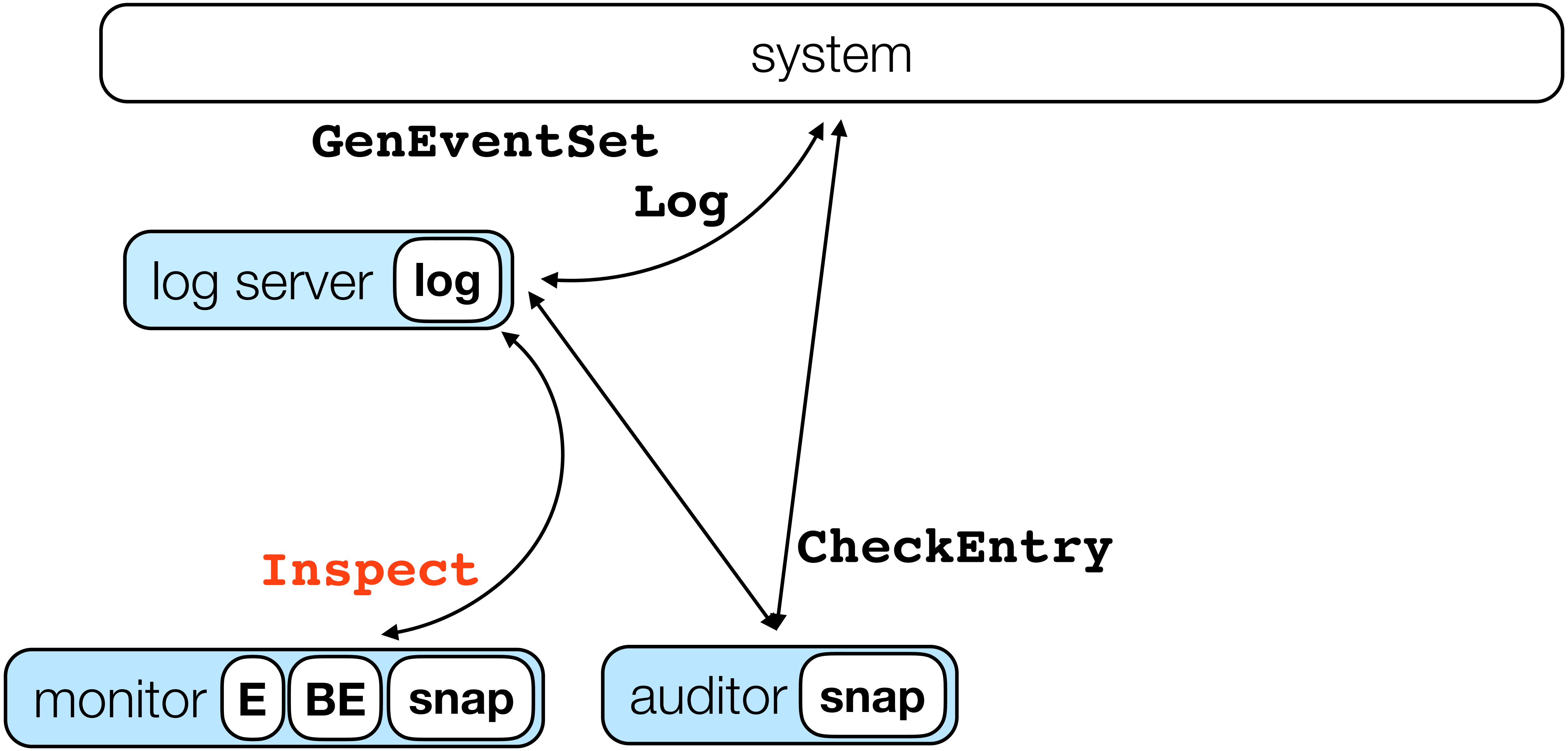
log server **log**

`CheckEntry`

auditor **snap**

(meaning |snap| ≪ |log|)

auditors (efficiently) determine if events are in the log



system

GenEventSet

Log

log server

log

CheckEntry

Inspect

monitor

E

BE

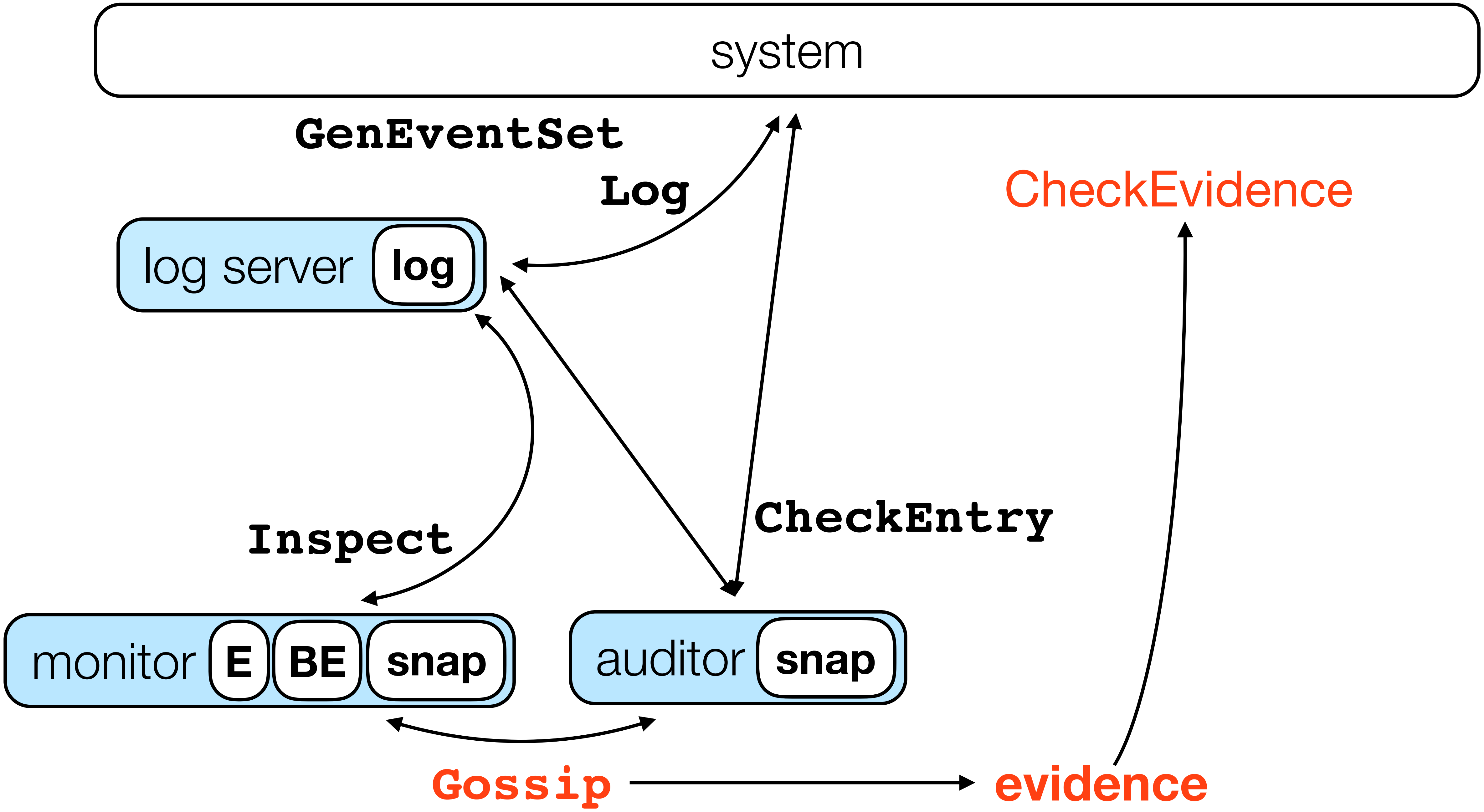
snap

auditor

snap

(meaning |E| ≈ |log|)

monitors (inefficiently) detect bad events in the log



system

GenEventSet

Log

CheckEvidence

log server **log**

Inspect

CheckEntry

monitor **E** **BE** **snap**

auditor **snap**

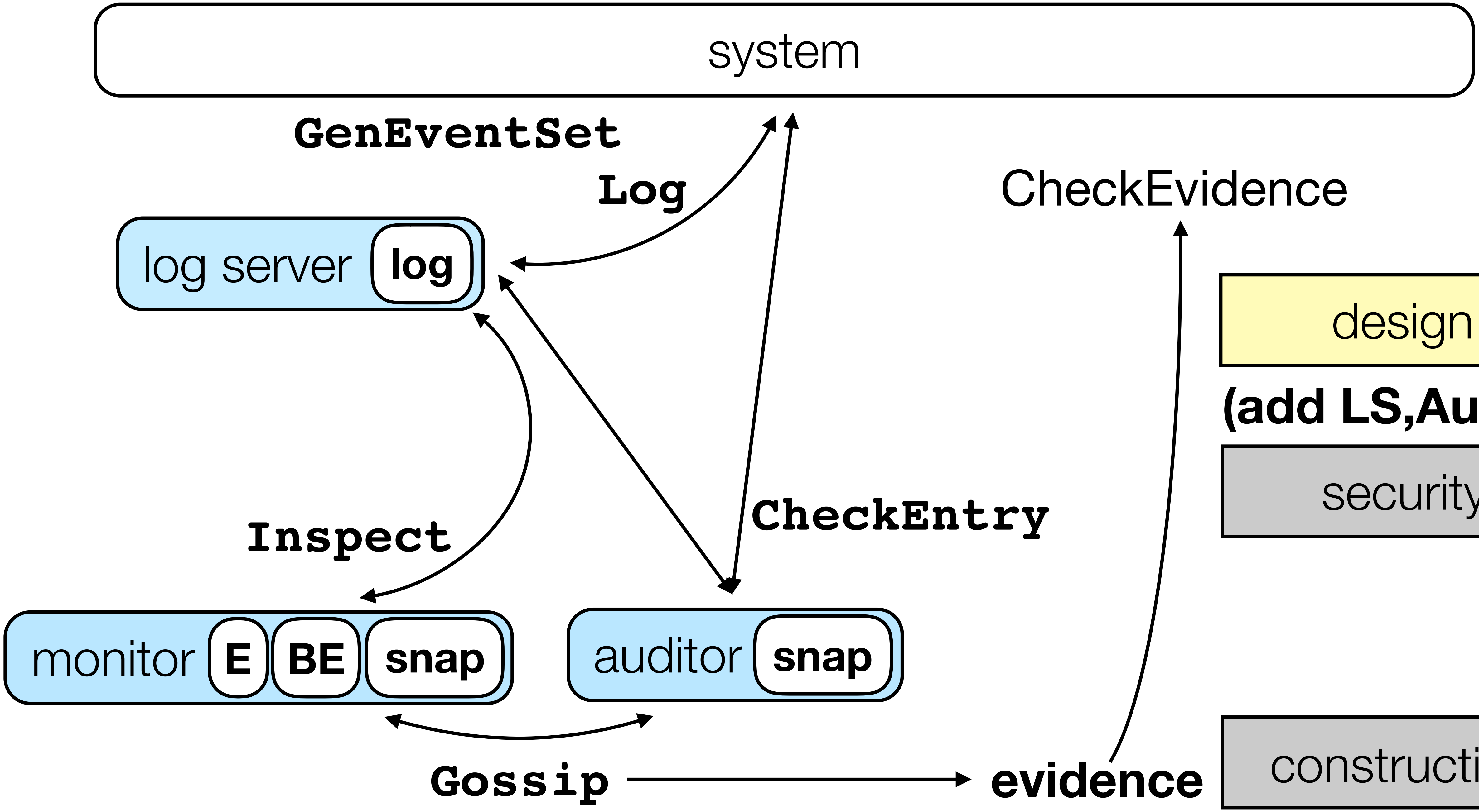
Gossip

evidence

auditors and monitors ensure consistent view of log

(can output evidence of inconsistencies)

which systems?



system

GenEventSet

Log

CheckEvidence

log server log

Inspect

CheckEntry

monitor E BE snap

auditor snap

Gossip

evidence

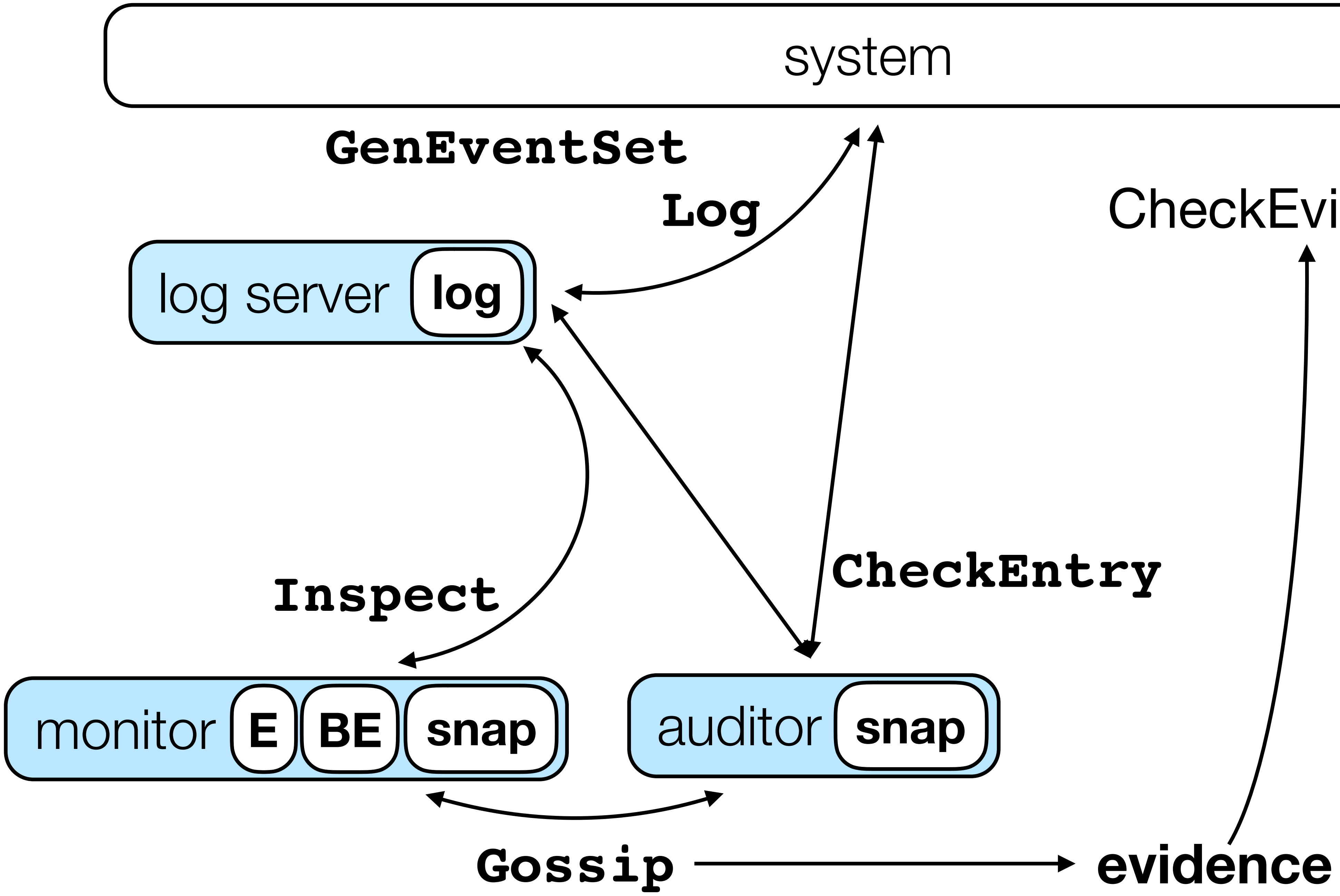
design

(add LS,Au,Mo)

security

construction

which systems?



system

GenEventSet

Log

CheckEvidence

log server

log

CheckEntry

Inspect

monitor

E

BE

snap

auditor

snap

Gossip

evidence

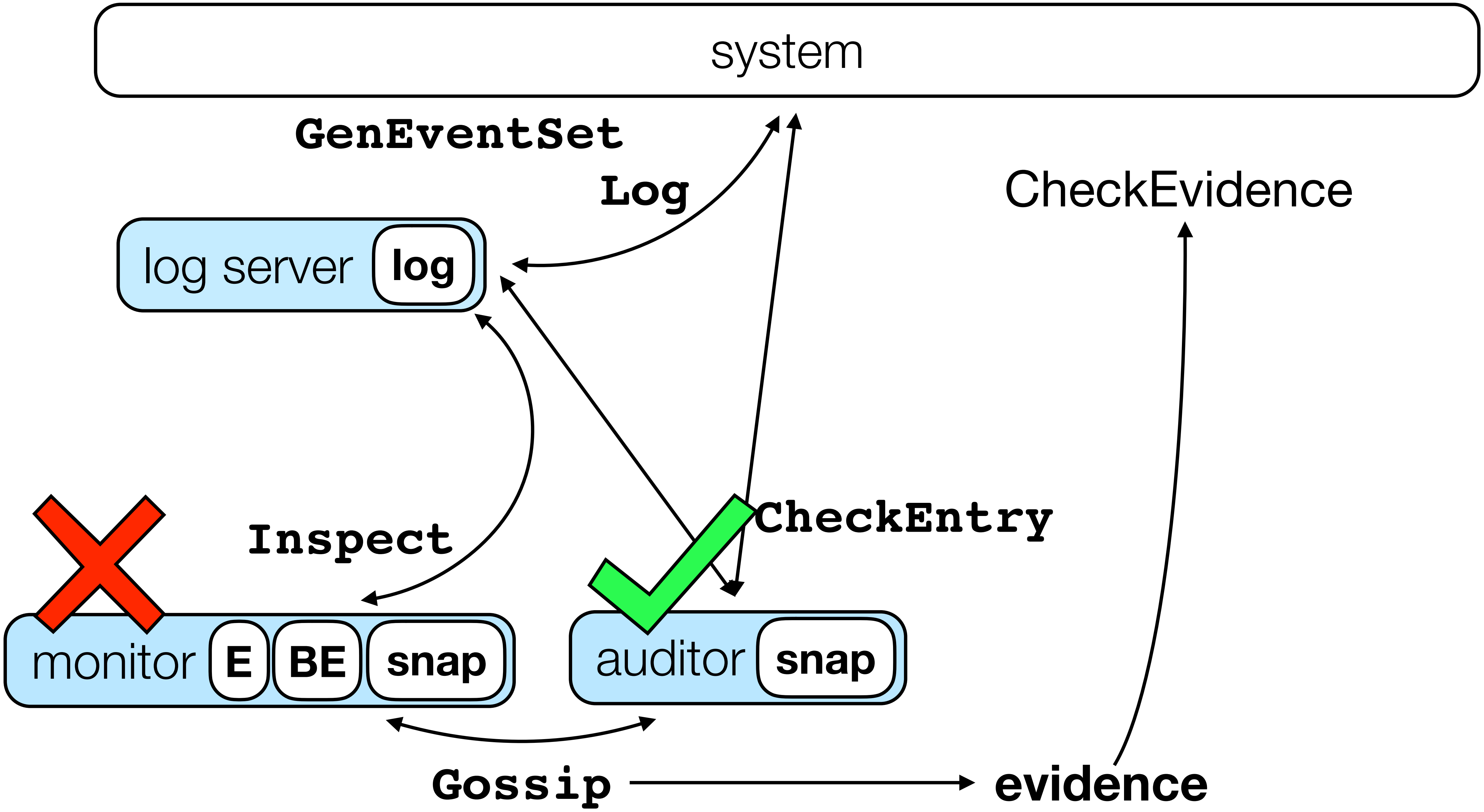
design

(add LS,Au,Mo)

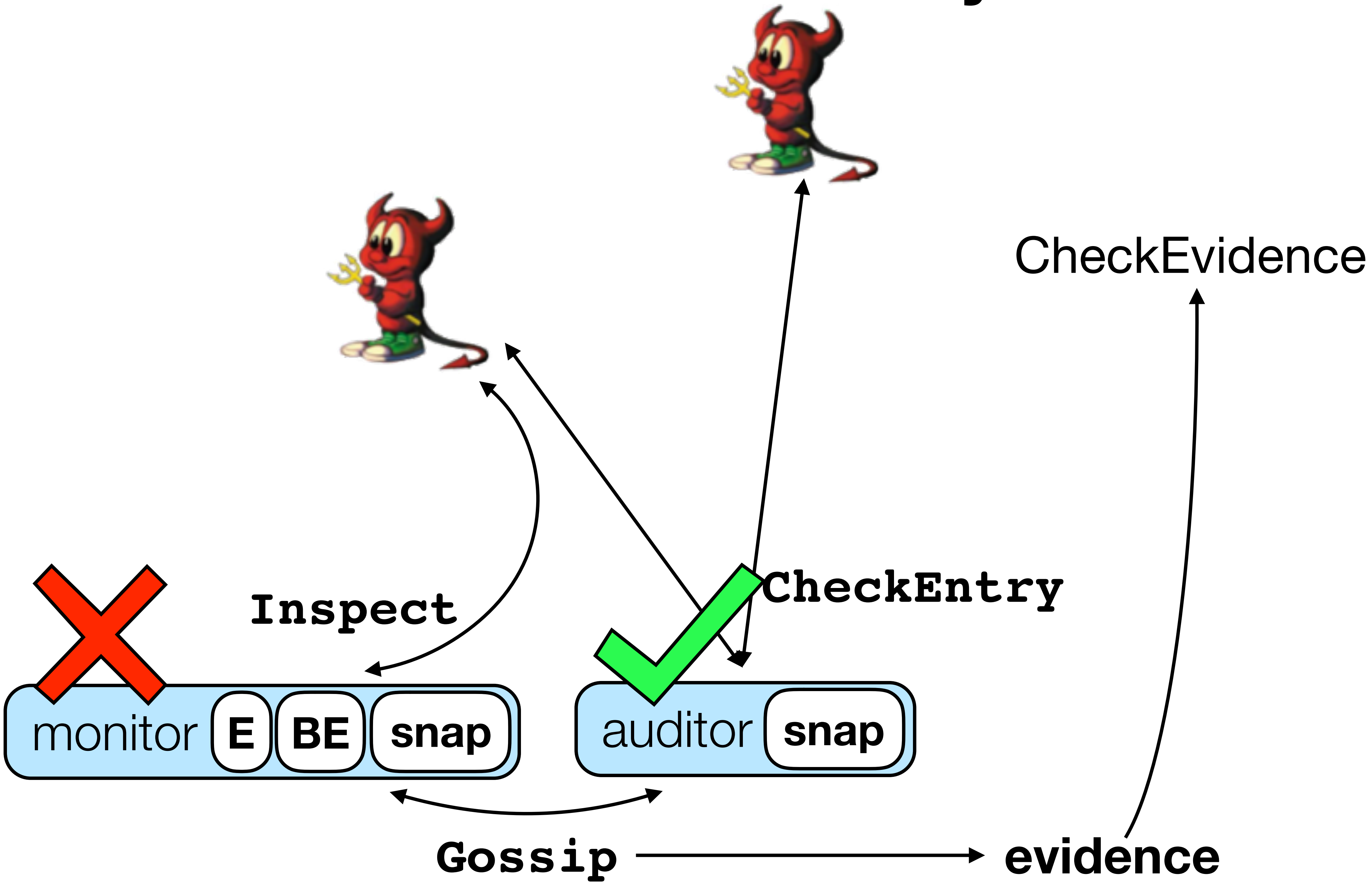
security

construction

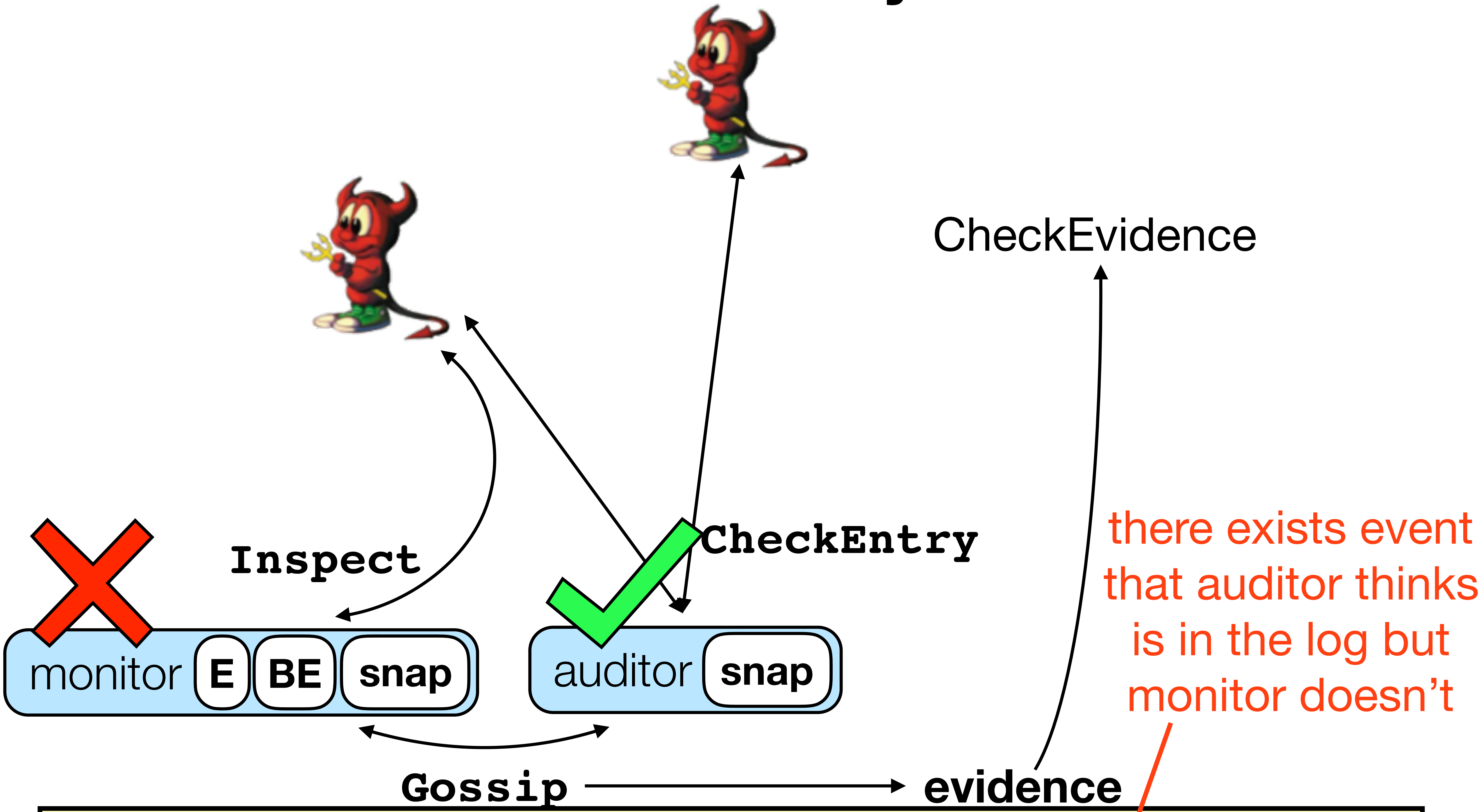
consistency



consistency

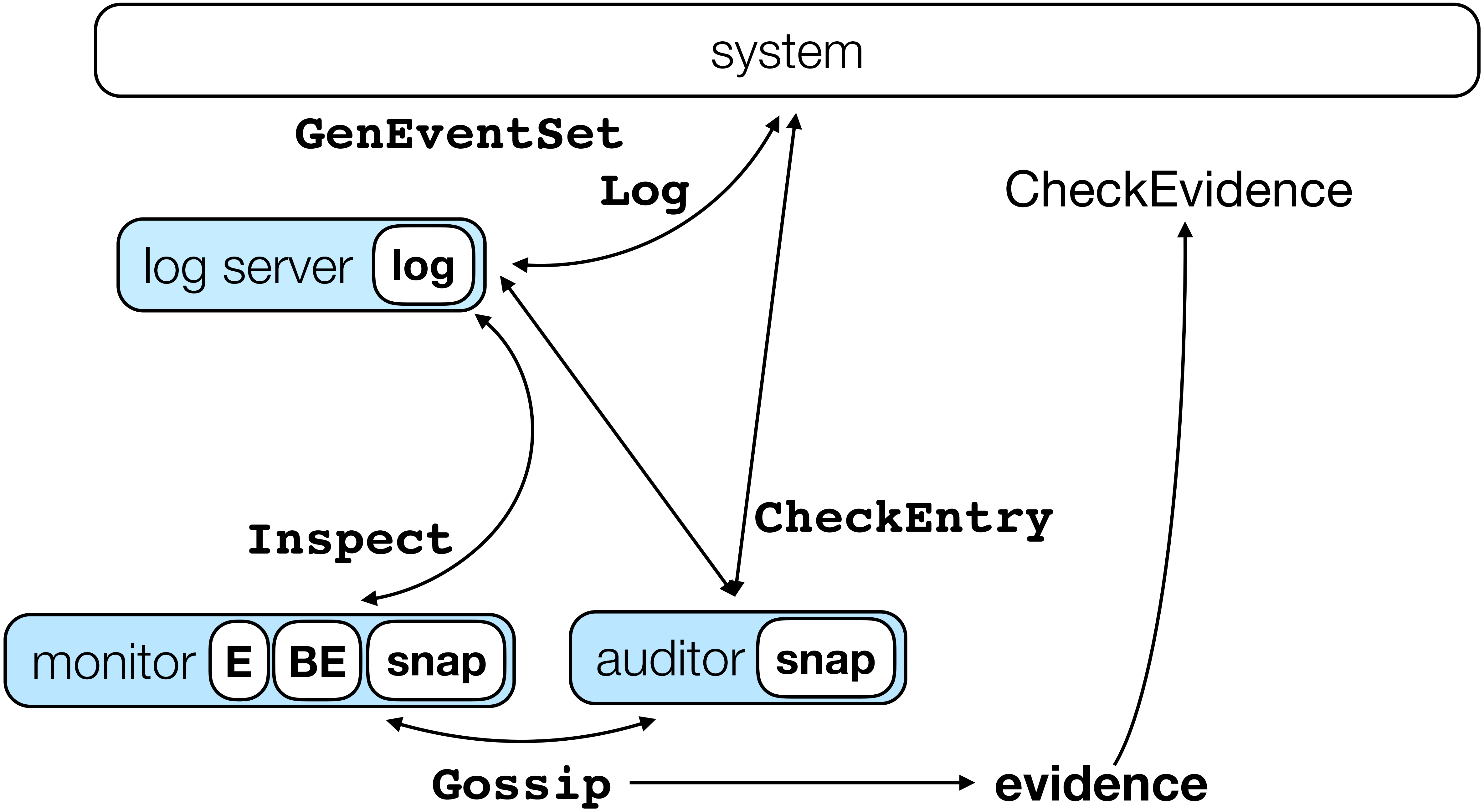


consistency

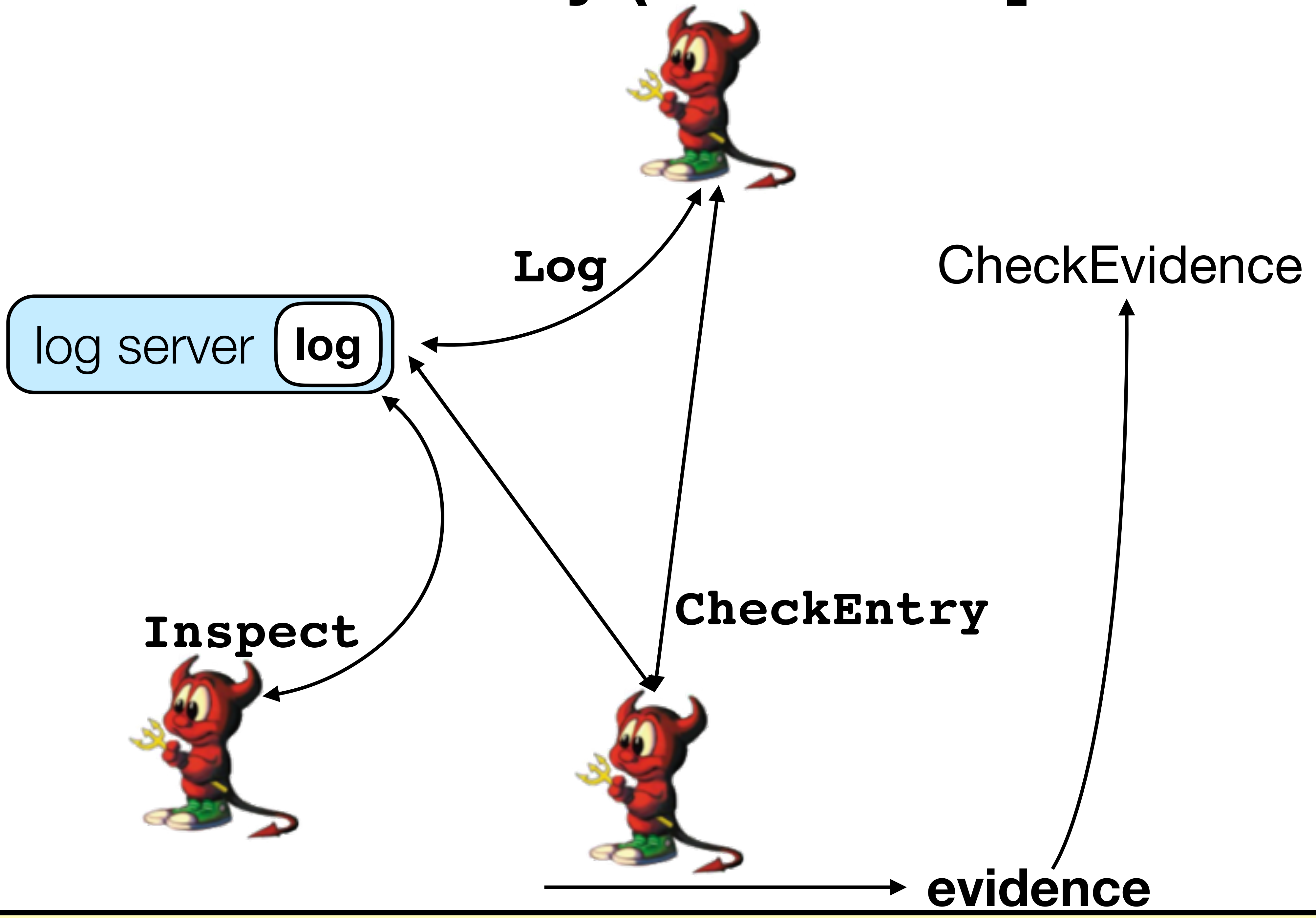

CheckEvidence
Inspect
CheckEntry
monitor
E
BE
snap
auditor
snap
Gossip
evidence
there exists event that auditor thinks is in the log but monitor doesn't

adversary wins if (1) **evidence fails** even though (2) monitor and auditor did have **inconsistent view**

non-frameability (related to [DGHS’16])

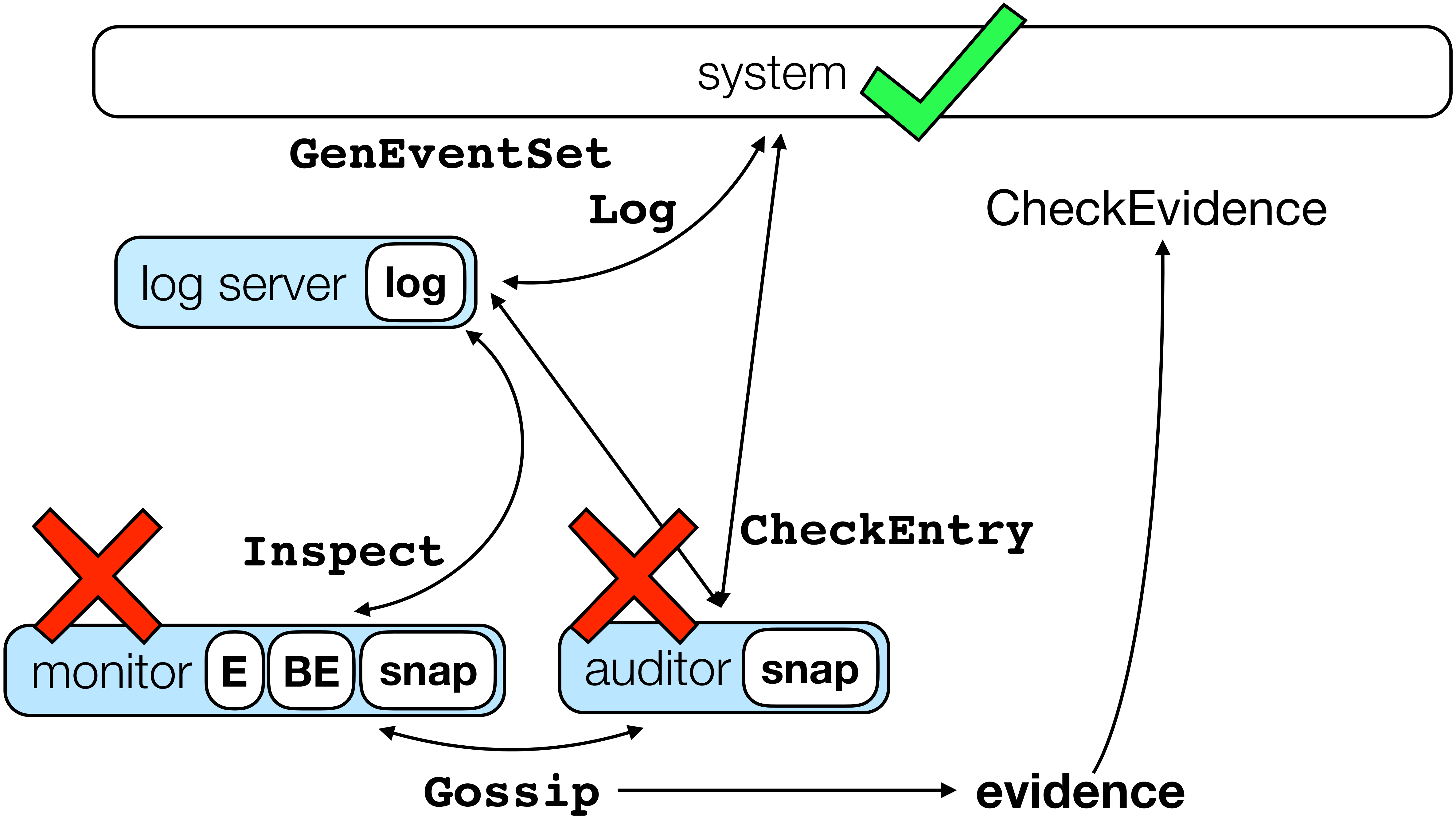


non-frameability (related to [DGHS'16])

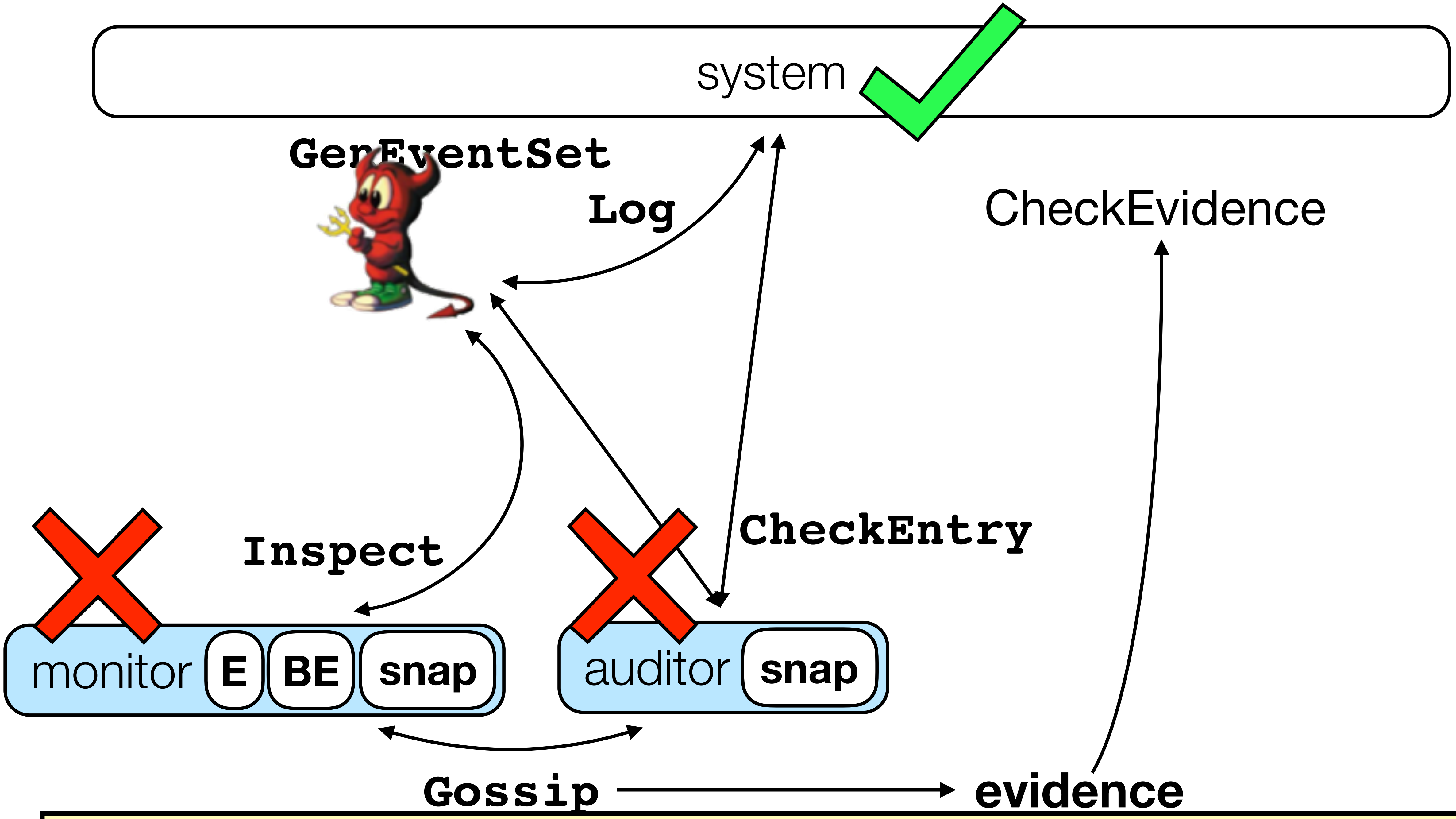


adversary wins if evidence passes

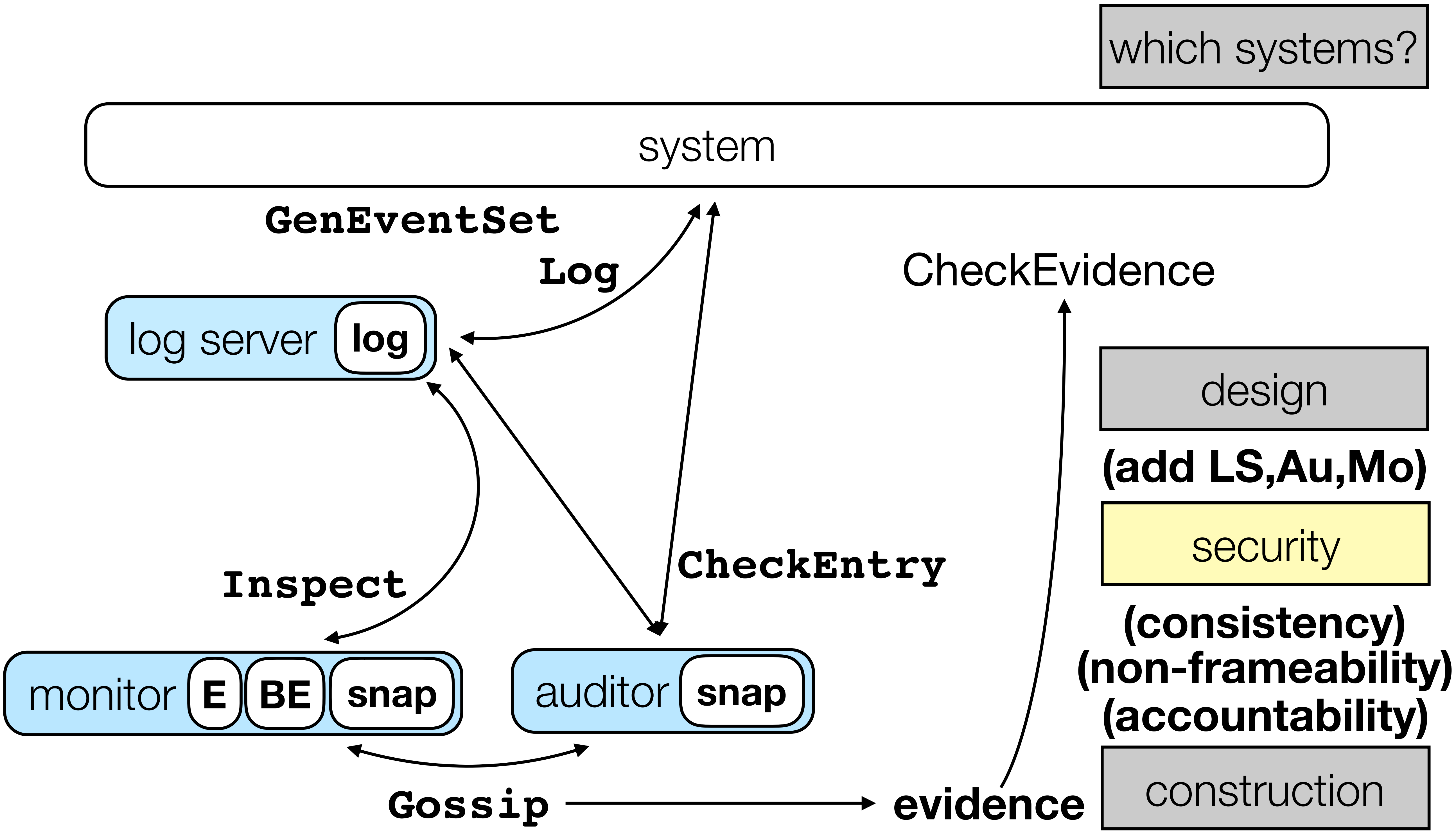
accountability



accountability



adversary wins if (1) **it promised** to include an event that (2) auditor and monitor believe to **not be in the log**, but (3) **evidence fails**



which systems?

system

GenEventSet

Log

CheckEvidence

log server log

design

(add LS,Au,Mo)

security

CheckEntry

Inspect

(consistency)
(non-frameability)
(accountability)

monitor E BE snap

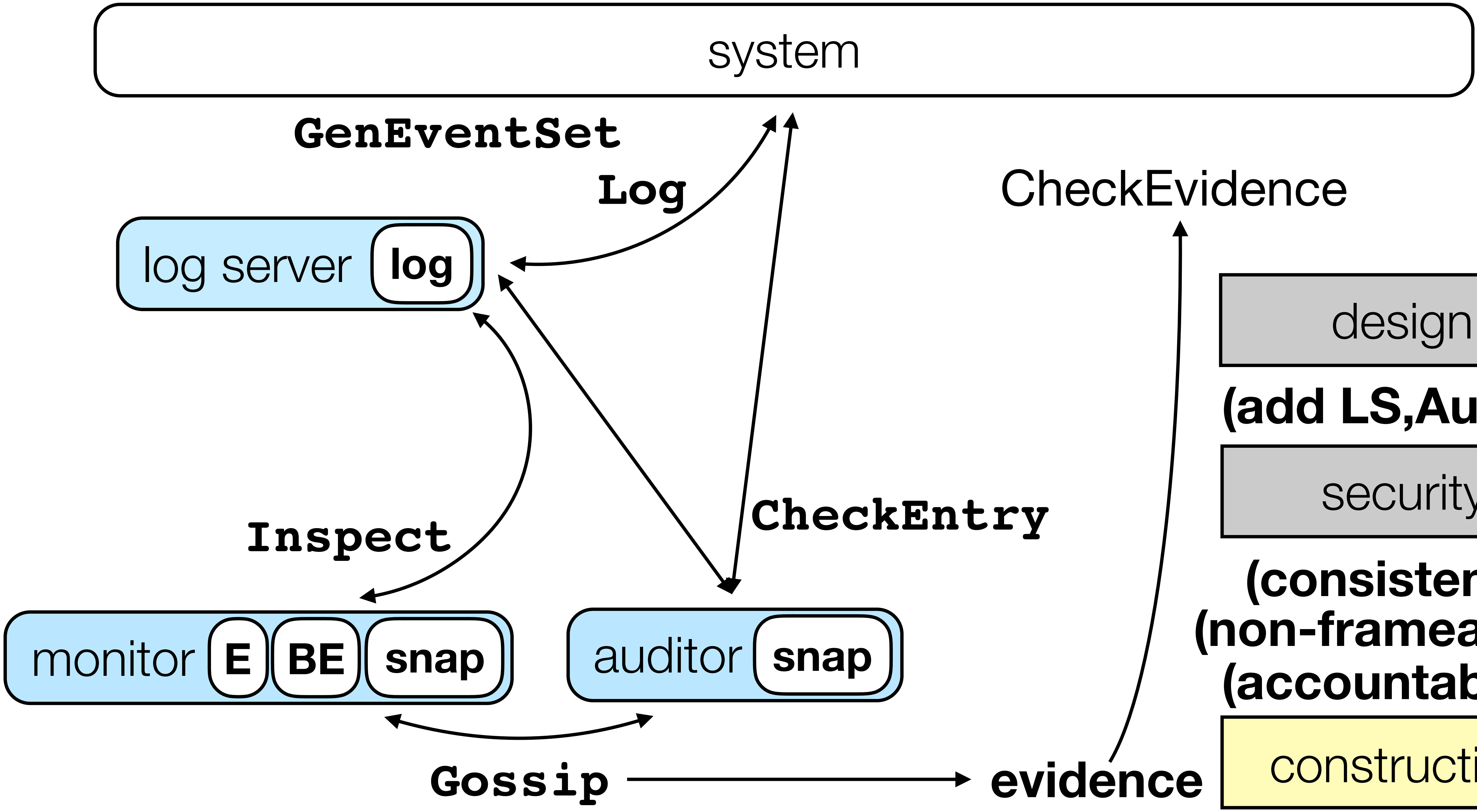
auditor snap

Gossip

evidence

construction

which systems?



system

GenEventSet

Log

CheckEvidence

log server **log**

Inspect

CheckEntry

monitor **E** **BE** **snap**

auditor **snap**

Gossip

evidence

design

(add LS,Au,Mo)

security

(consistency)
(non-frameability)
(accountability)

construction

dynamic list commitment (dlc)

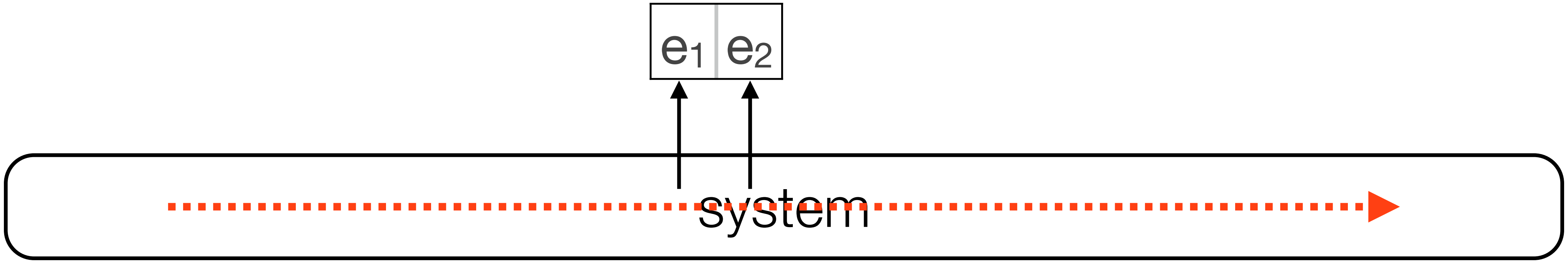
(aka tamper-evident log [CW’09])

(aka authenticated data structure [AGT’01,PSTY’13])

(aka rolling hash chain or Merkle tree [M’89])

dynamic list commitment (dlc)

dynamic list commitment (dlc)



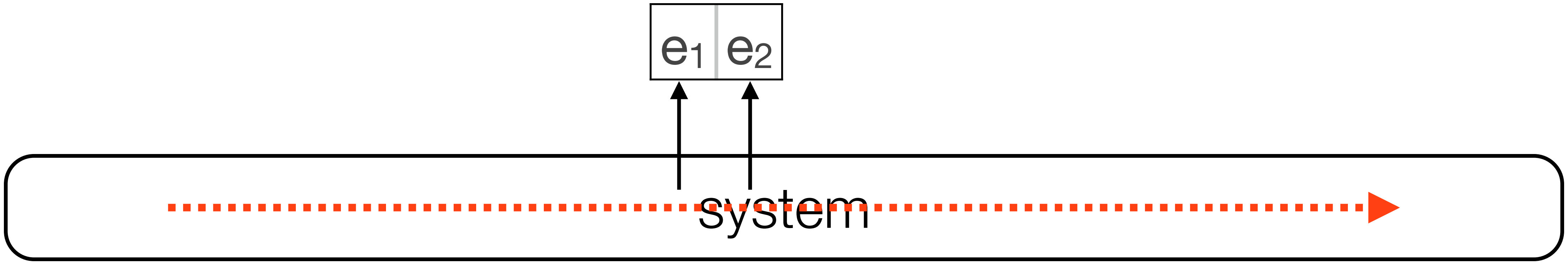
dynamic list commitment (dlc)

basic

Com

CheckCom

Append



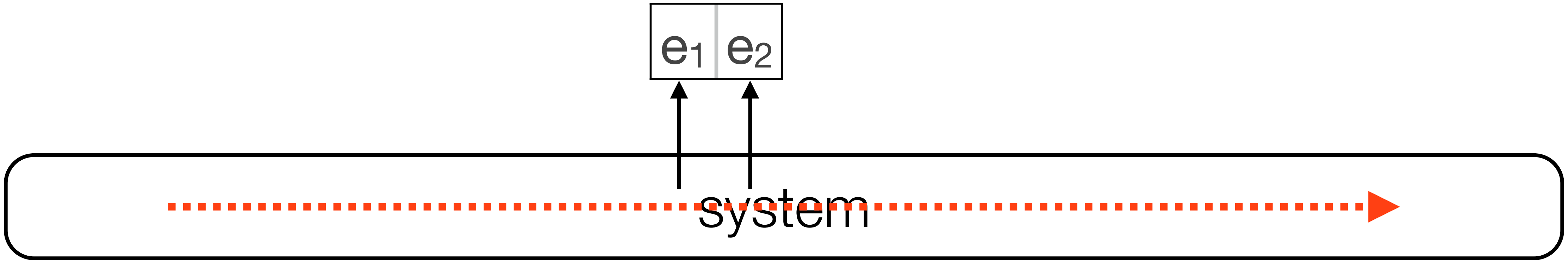
dynamic list commitment (dlc)

basic

Com (generate succinct commitment)

CheckCom

Append



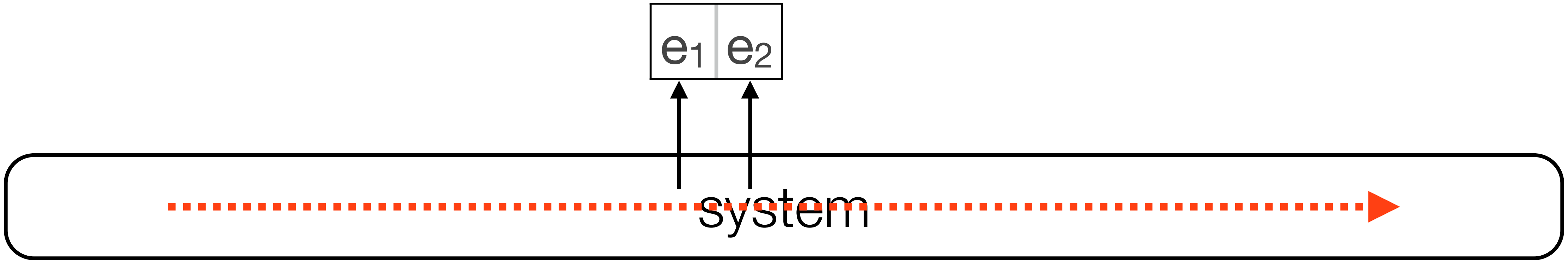
dynamic list commitment (dlc)

basic

Com (generate succinct commitment)

CheckCom (check commitment)

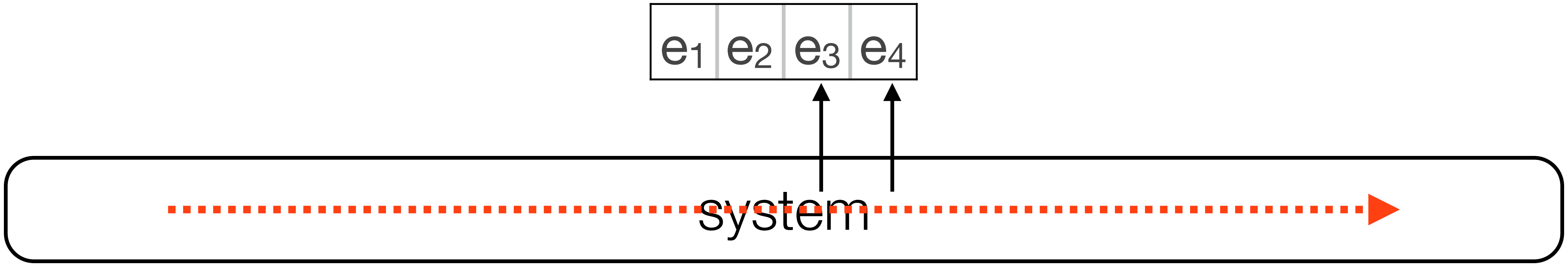
Append



dynamic list commitment (dlc)

basic

Com	(generate succinct commitment)
CheckCom	(check commitment)
Append	(add new events)



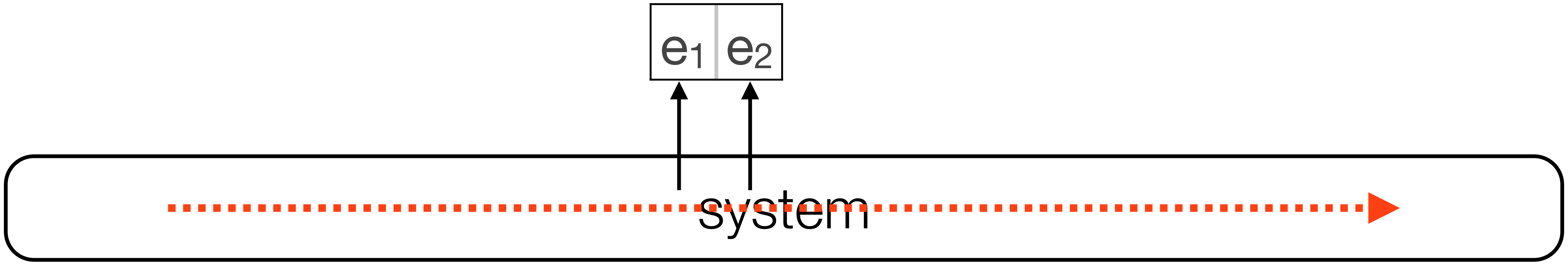
dynamic list commitment (dlc)

basic

Com
CheckCom
Append

all events?

ProveAppend
CheckAppend



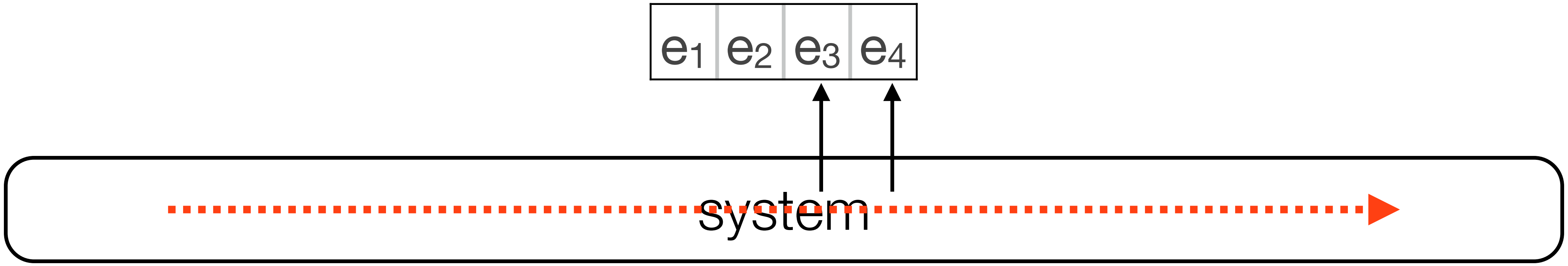
dynamic list commitment (dlc)

basic

Com
CheckCom
Append

all events?

ProveAppend
CheckAppend
(can’t delete events)



dynamic list commitment (dlc)

basic

Com
CheckCom
Append

all events?

ProveAppend
CheckAppend

specific event?

ProveIncl
CheckIncl
(can’t omit events)

e_1	e_2	e_3	e_4

dynamic list commitment (dlc)

basic

Com

CheckCom

Append

all events?

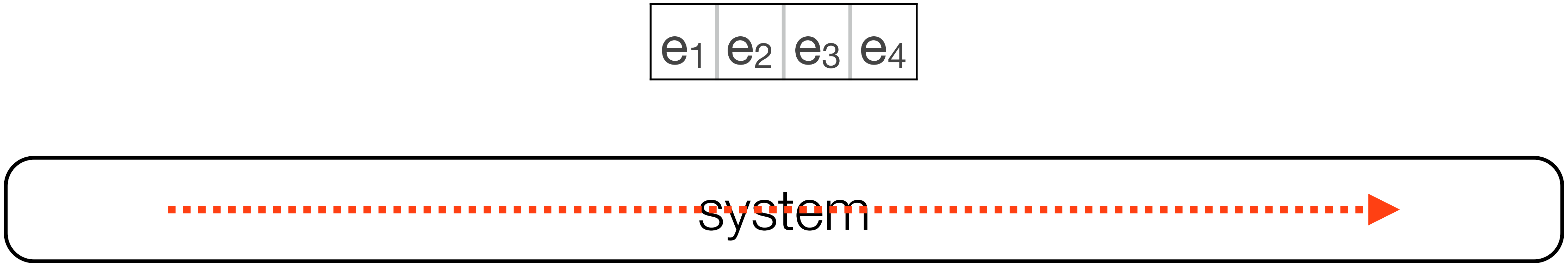
ProveAppend

CheckAppend

specific event?

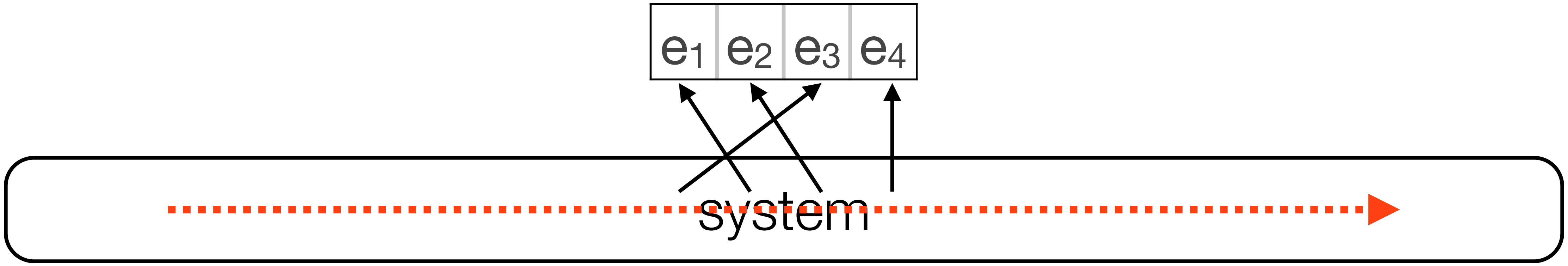
ProveIncl

CheckIncl



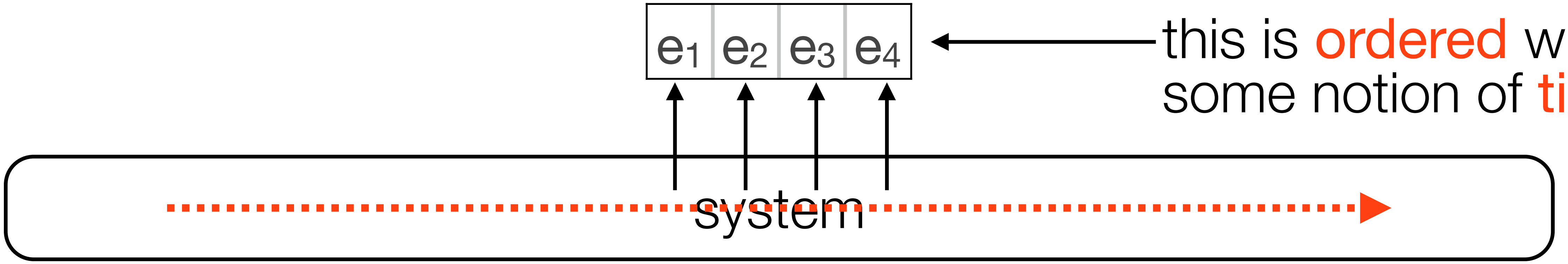
dynamic list commitment (dlc)

basic	all events?	specific event?
Com CheckCom Append	ProveAppend CheckAppend	ProveIncl CheckIncl



dynamic list commitment (dlc)

basic	all events?	specific event?
Com CheckCom Append	ProveAppend CheckAppend	ProveIncl CheckIncl



this is ordered w.r.t. some notion of time

dynamic list commitment (dlc)

basic

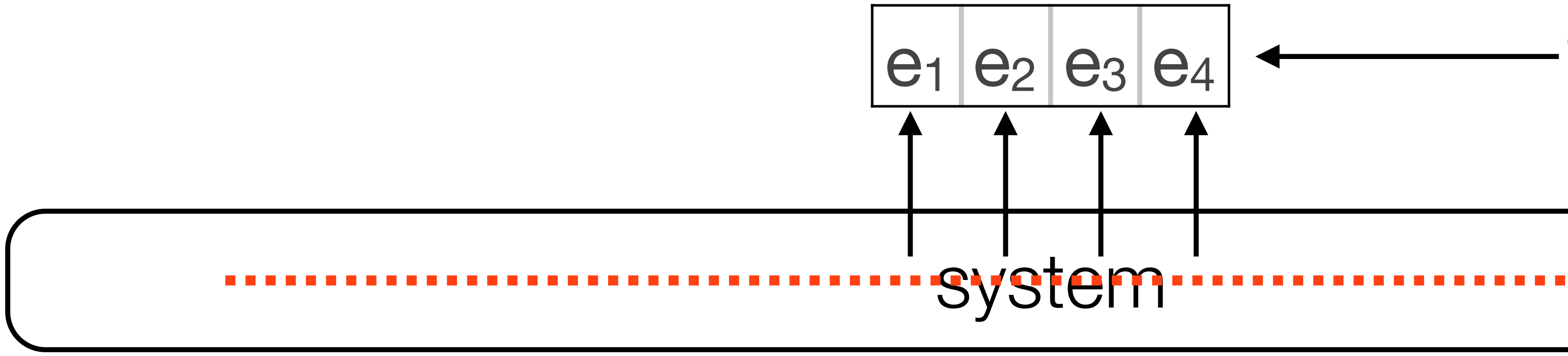
Com
CheckCom
Append

all events?

ProveAppend
CheckAppend

specific event?

ProveIncl
CheckIncl



this is ordered w.r.t.
some notion of time

inconsistent?

DemoInconsistent
CheckInconsistent

"your commitment c does not
represent the state of my list at time t"

dynamic list commitment (dlc)

basic

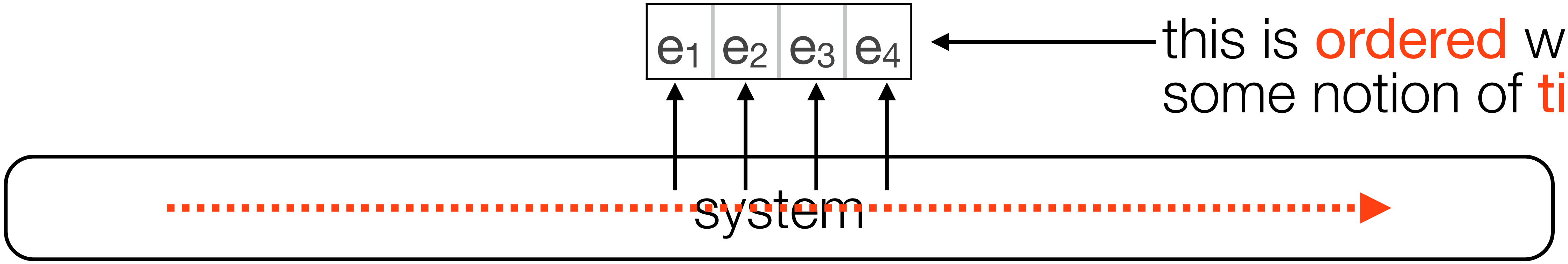
Com
CheckCom
Append

all events?

ProveAppend
CheckAppend

specific event?

ProveIncl
CheckIncl



this is ordered w.r.t. some notion of time

inconsistent?

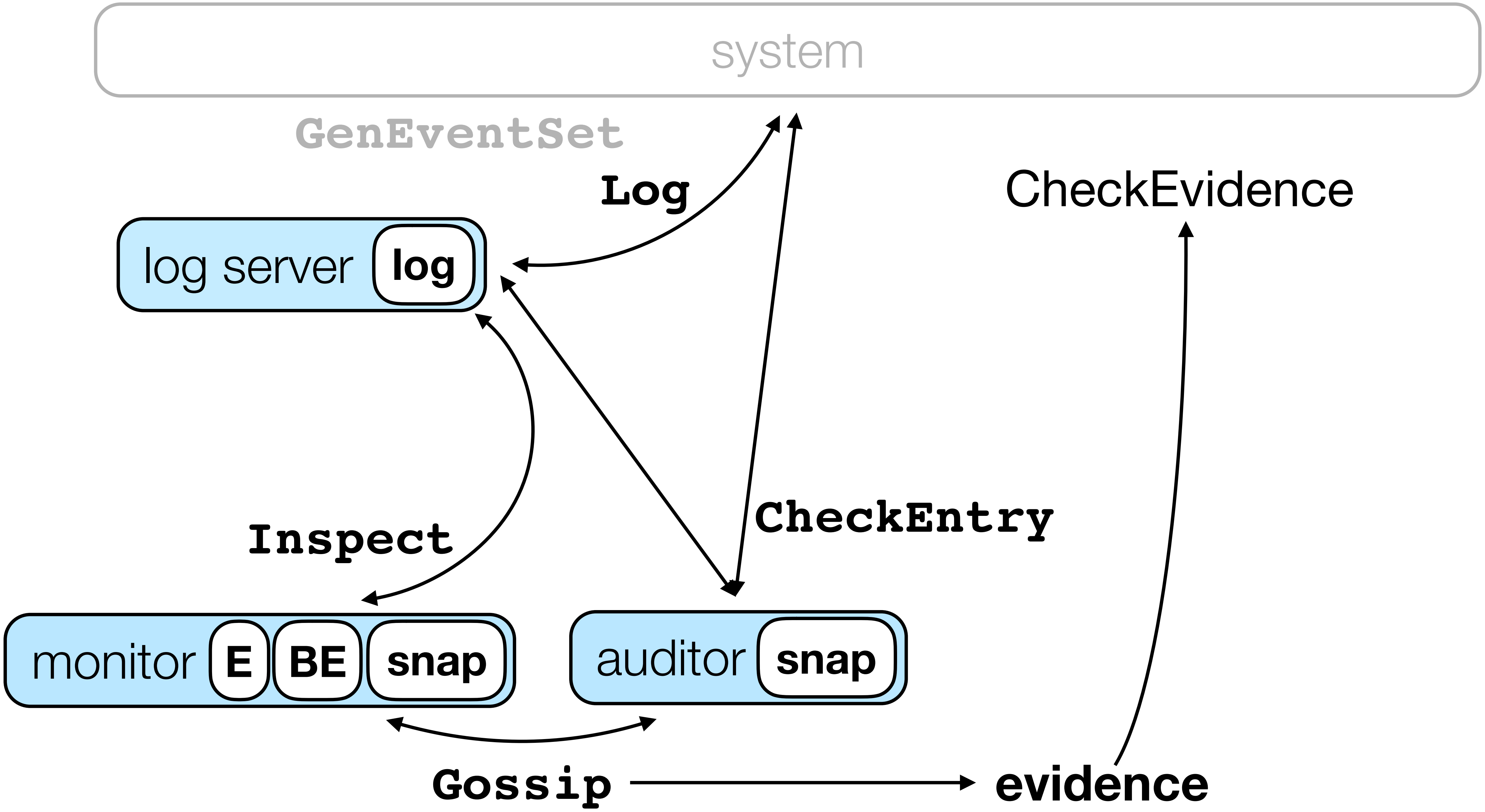
DemoInconsistent
CheckInconsistent

non-inclusion?

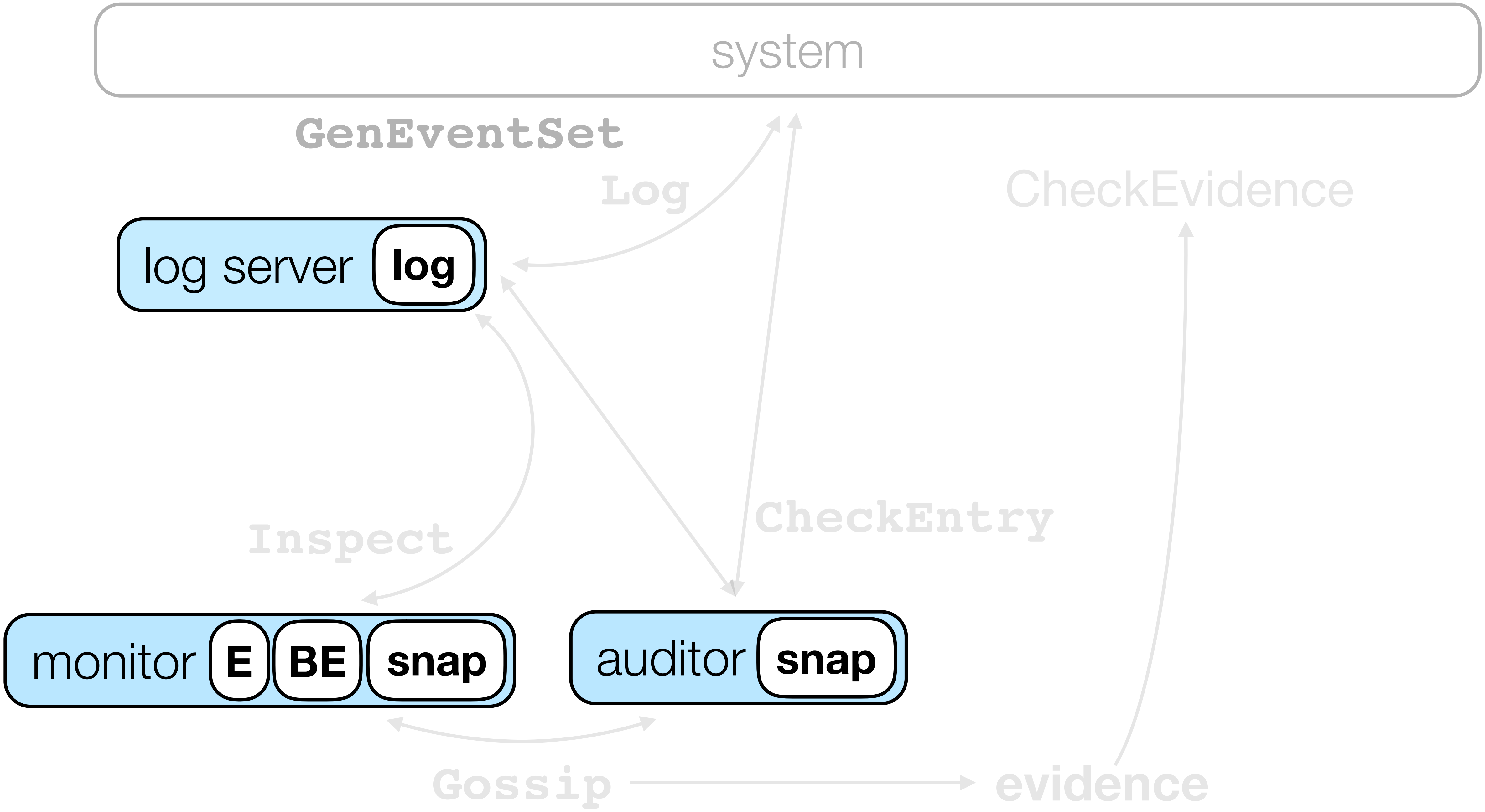
DemoNotIncl
CheckNotIncl

“your commitment c does not represent the state of my list at time t”

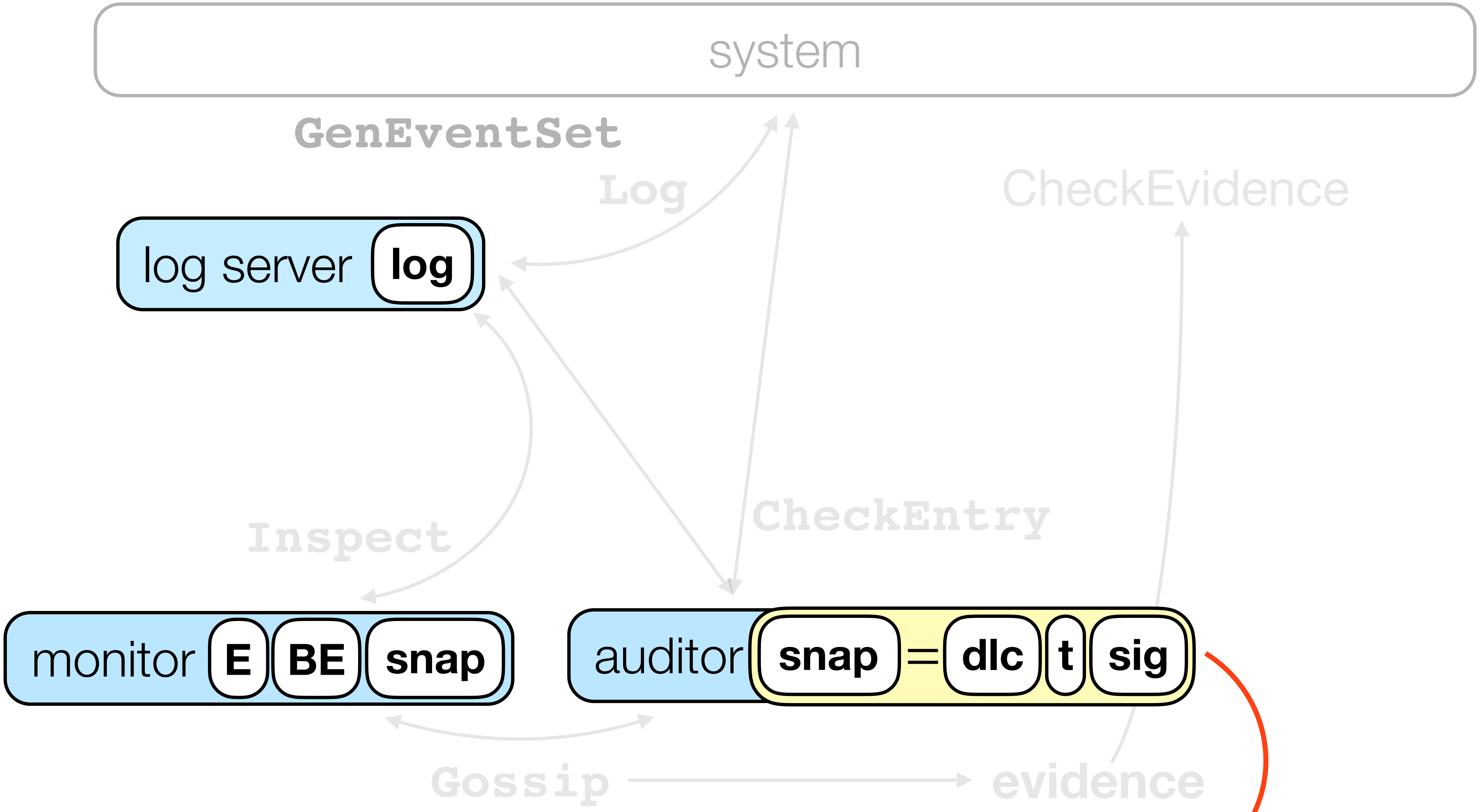
construction



construction

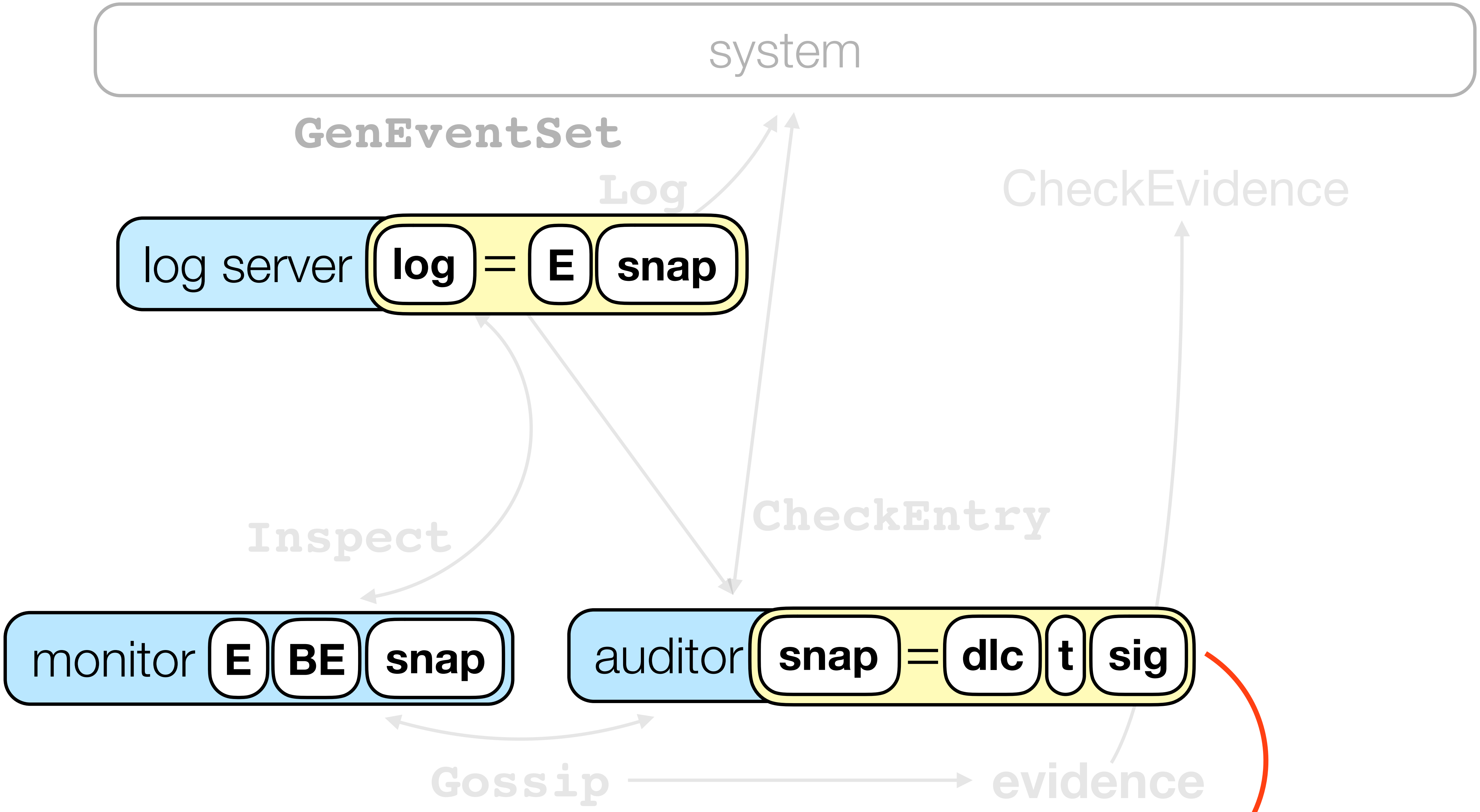


construction

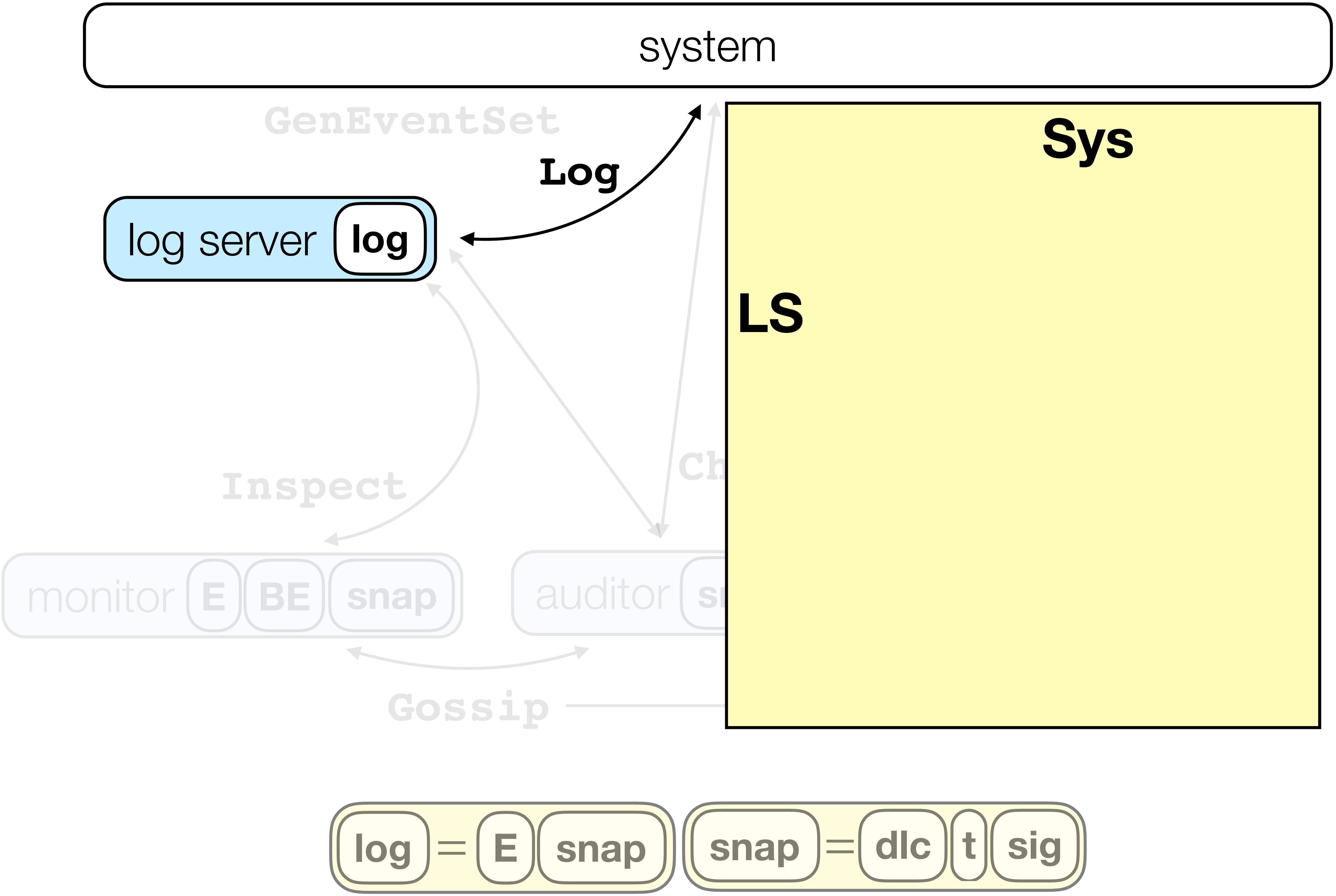


a (timed) signature, so
no one can frame LS

construction



a (timed) signature, so
no one can frame LS



system

GenEventSet

Log

log server log

Sys

LS

Inspect

Ch

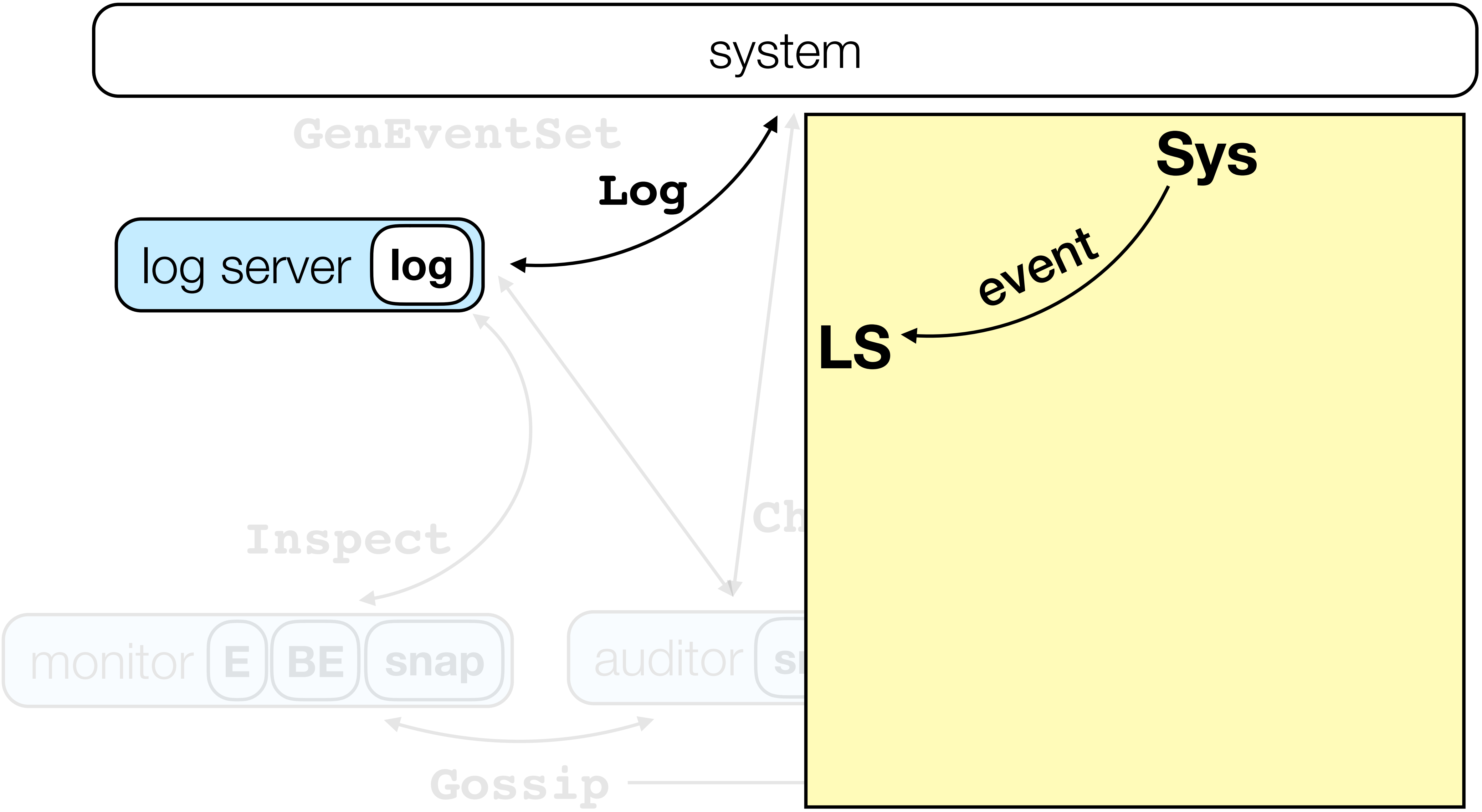
monitor E BE snap

auditor s

Gossip

log = E snap

snap = dlc t sig



system

GenEventSet

Log

log server log

Sys

event

LS

Inspect

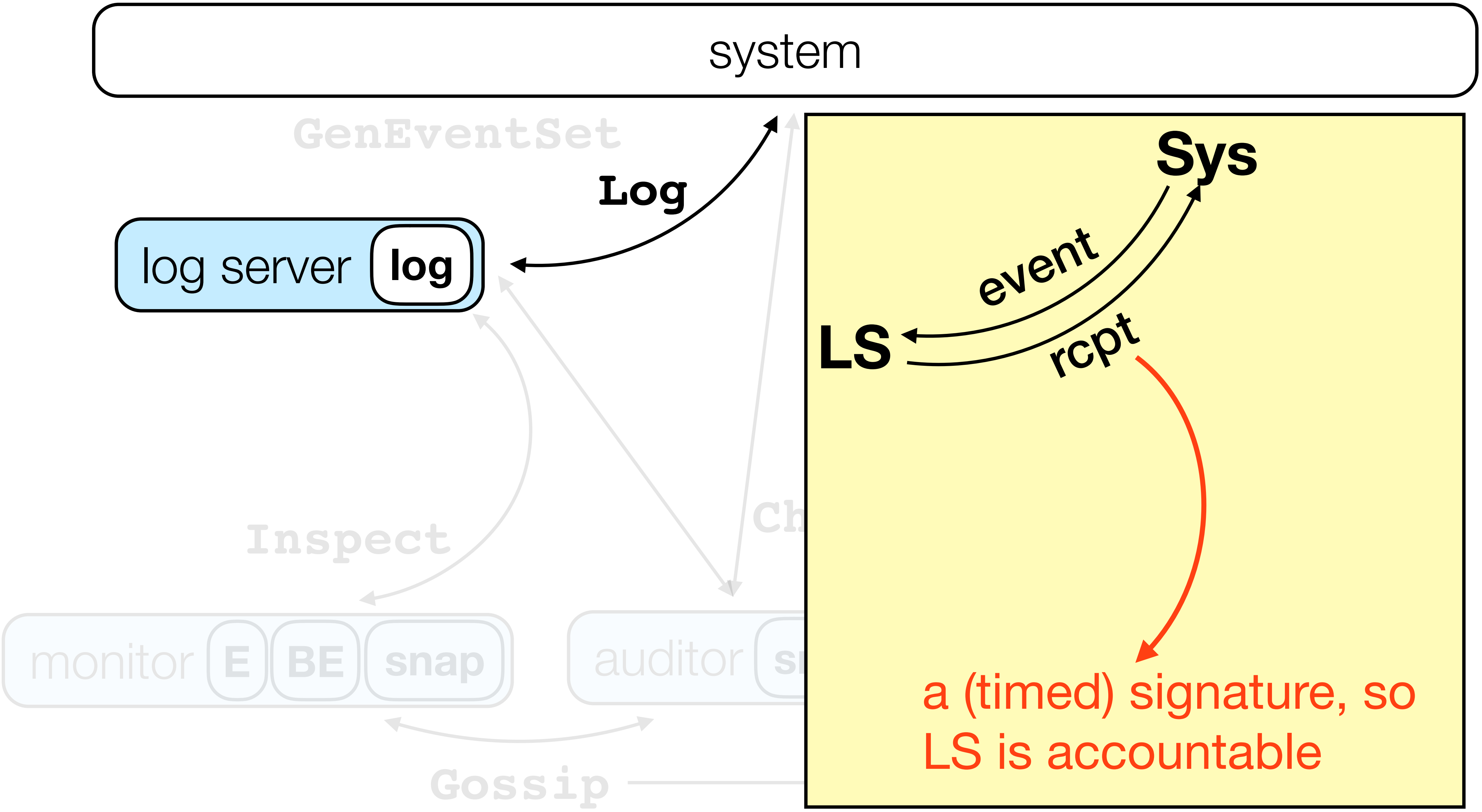
Ch

monitor E BE snap

auditor s

Gossip

log = E snap   snap = dlc t sig



system

GenEventSet

Log

log server log

Inspect

Ch

monitor E BE snap

auditor s

Gossip

Sys

event

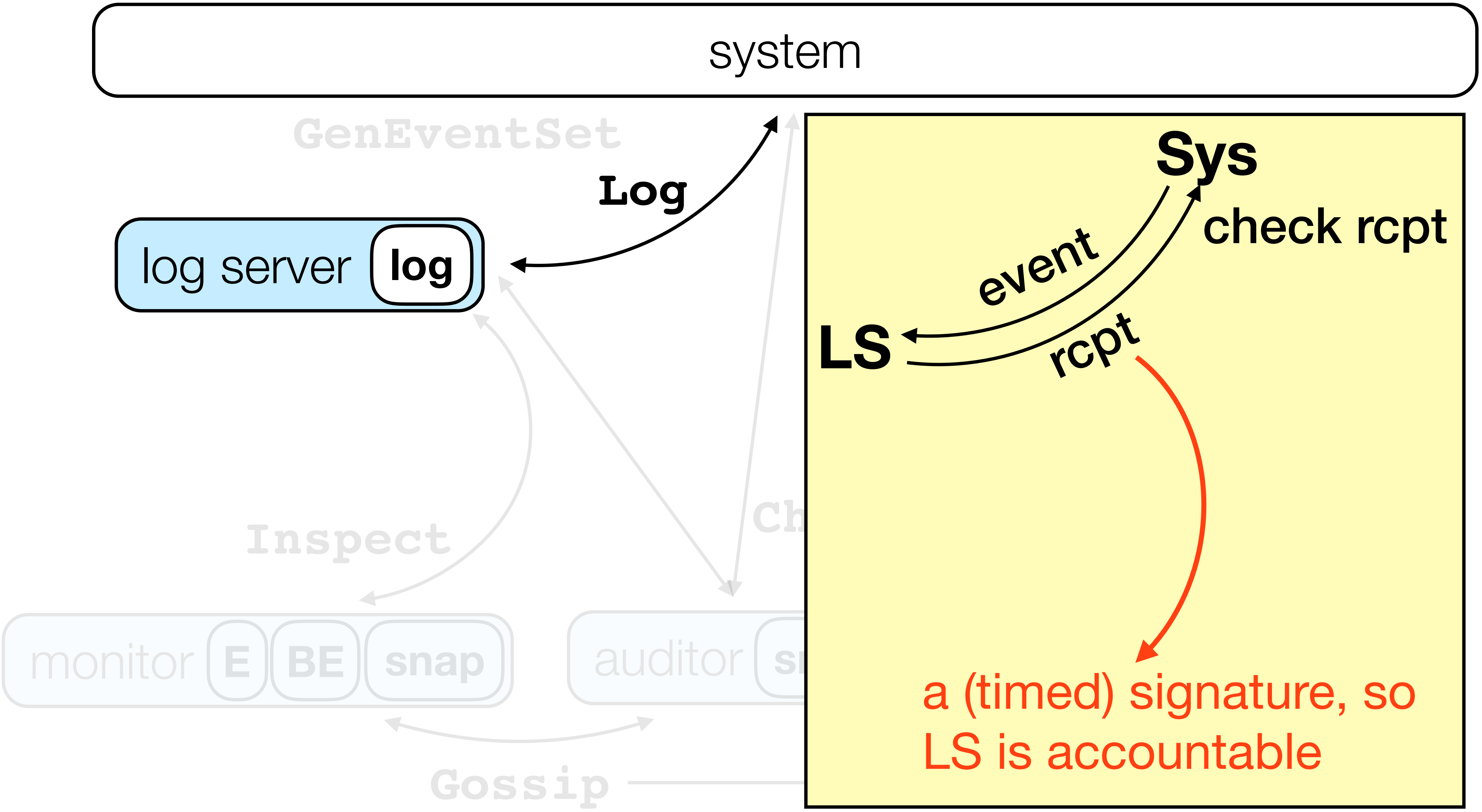
LS

rcpt

a (timed) signature, so
LS is accountable

log = E snap

snap = dlc t sig



system

GenEventSet

Log

log server log

Inspect

monitor E BE snap

auditor s

Ch

Gossip

Sys

check rcpt

event

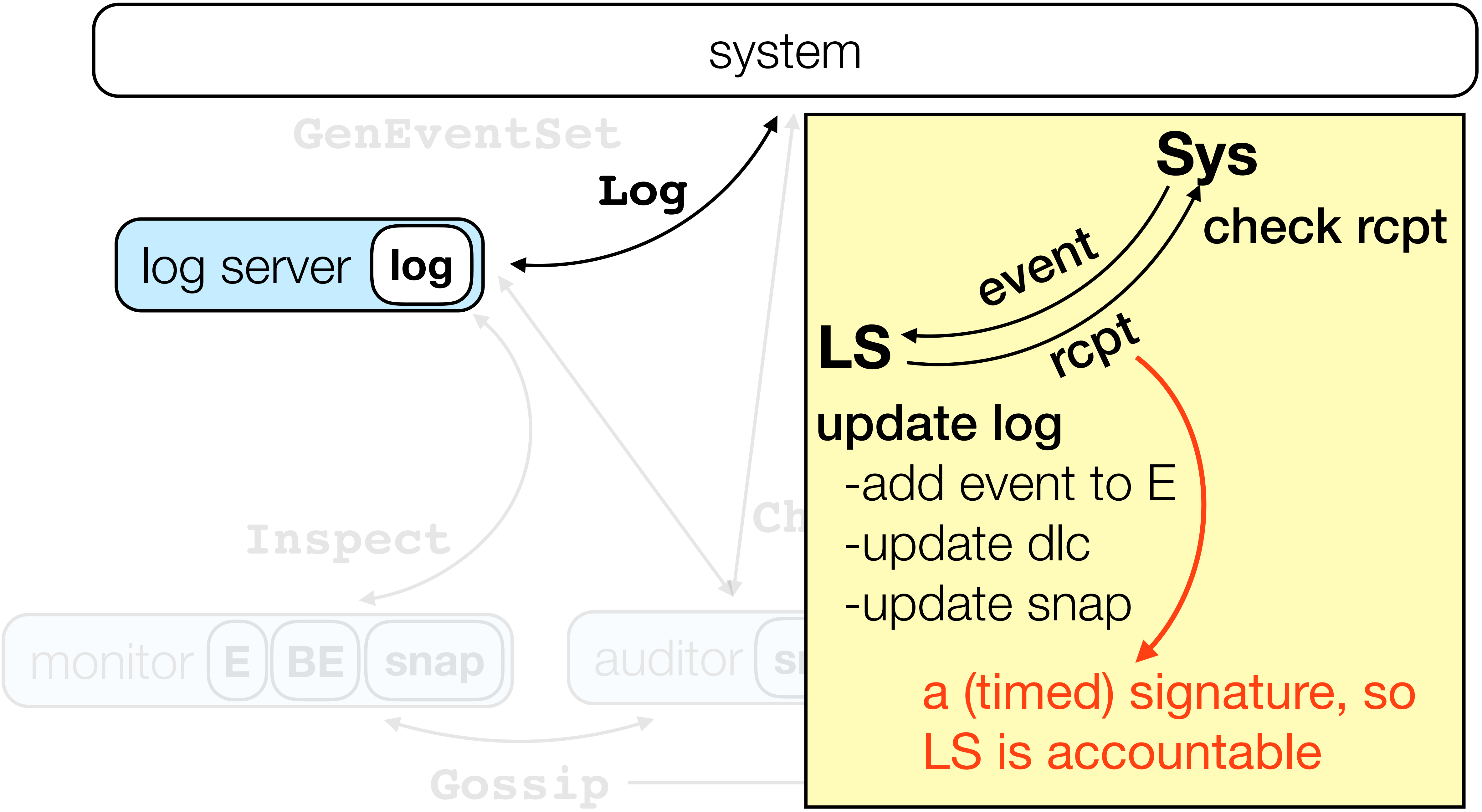
rcpt

LS

a (timed) signature, so LS is accountable

log = E snap

snap = dlc t sig



Sys

event

check rcpt

LS

rcpt

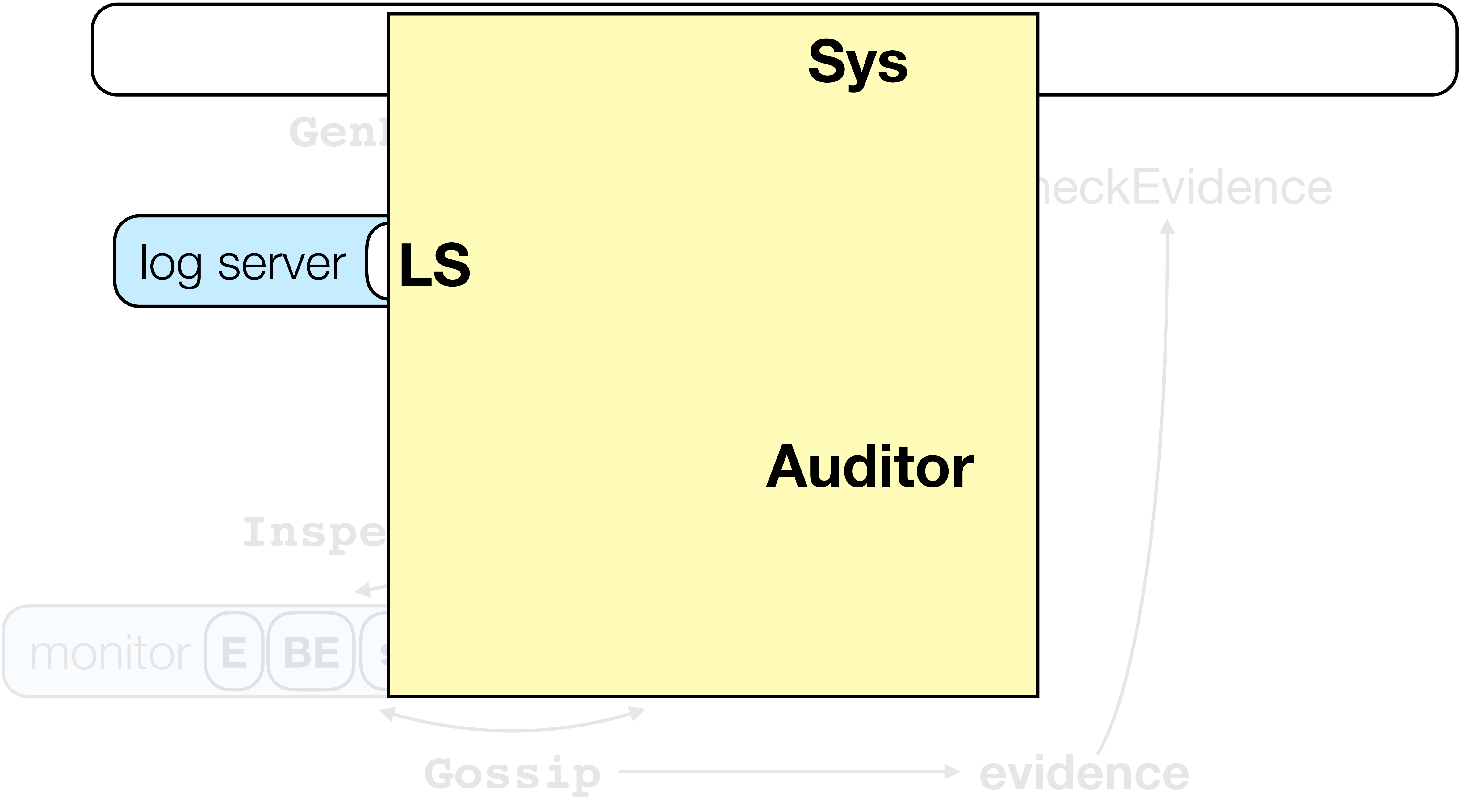
update log

- -add event to E
- -update dlc
- -update snap

a (timed) signature, so LS is accountable

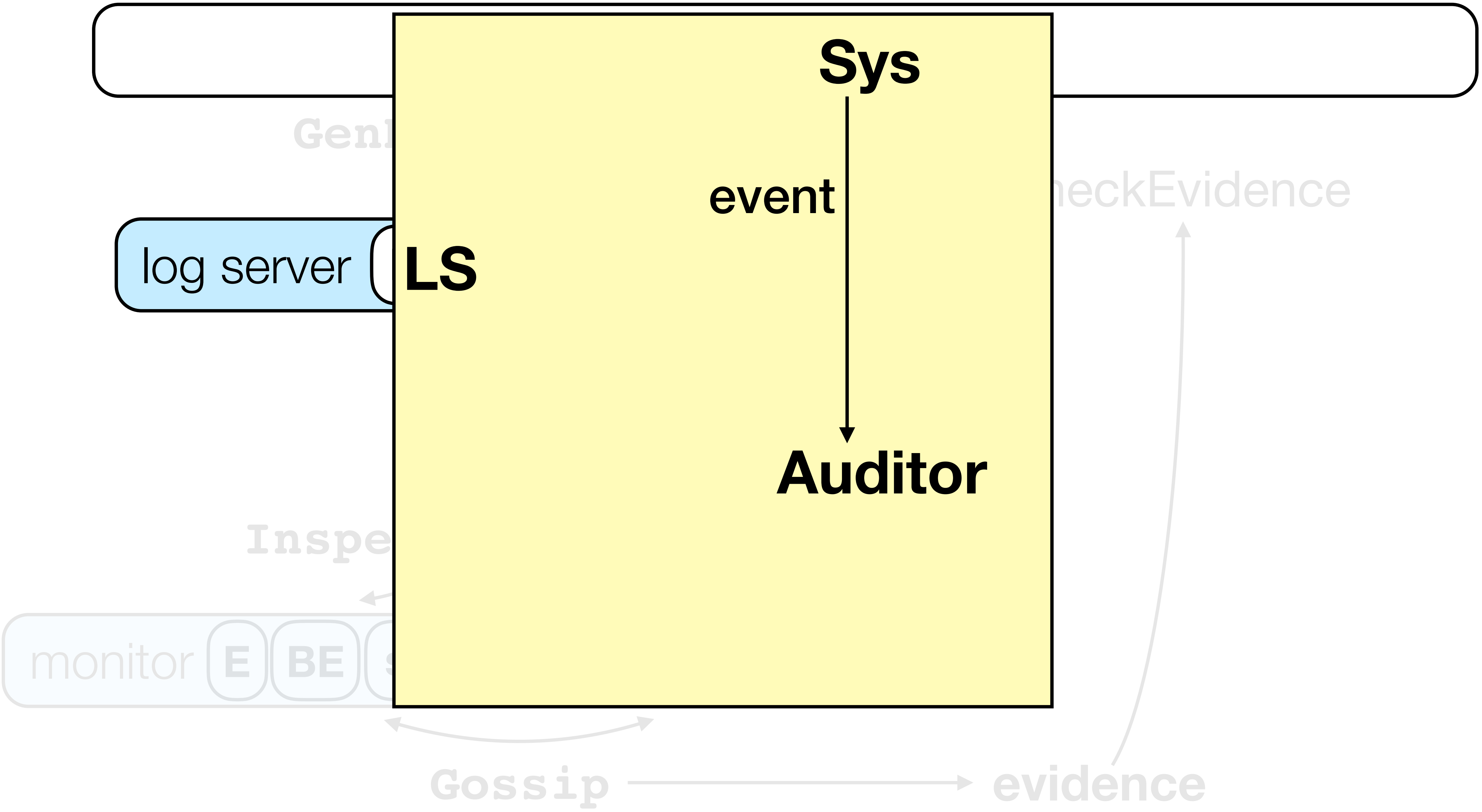
log = E snap

snap = dlc t sig



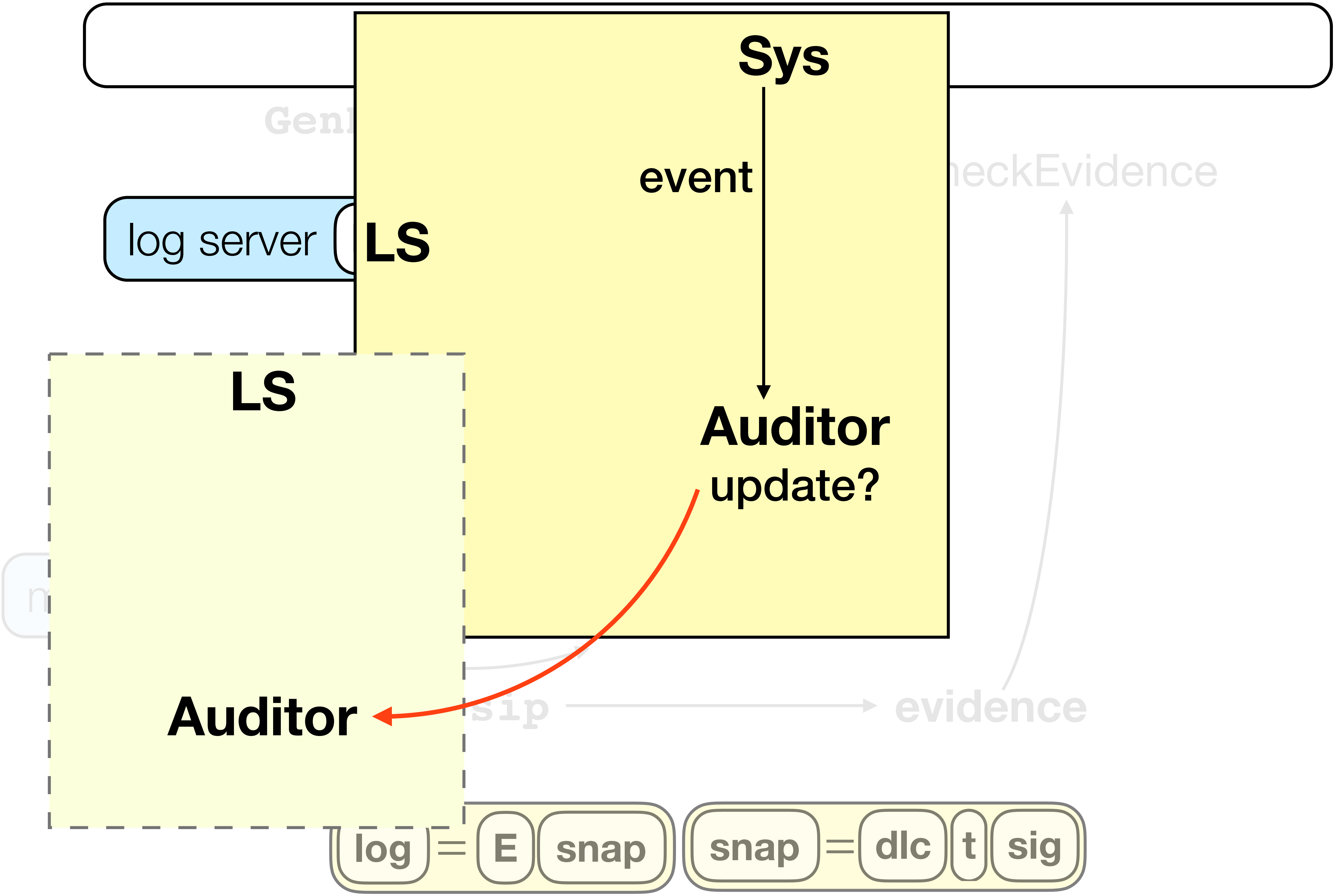
log = E snap

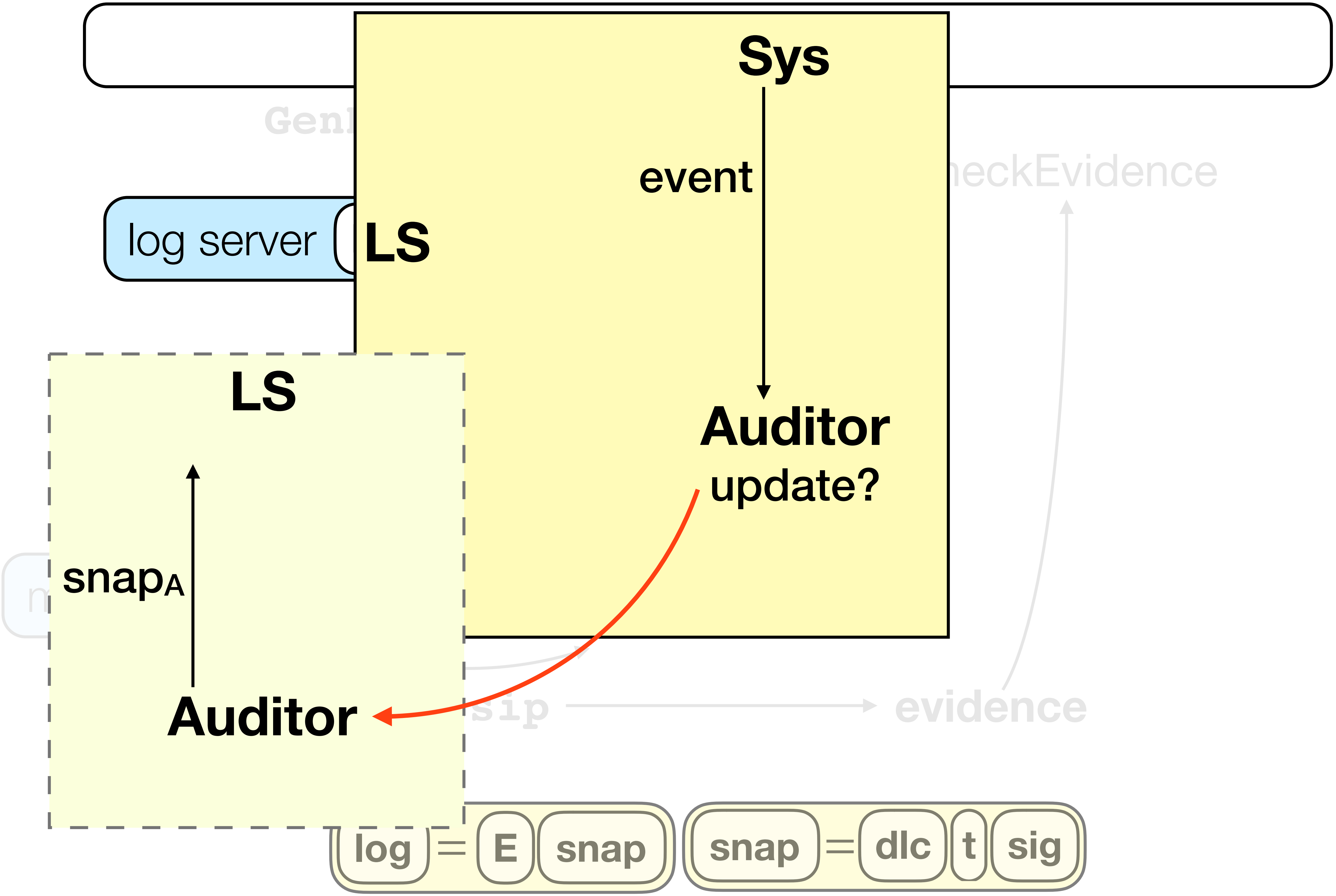
snap = dlc t sig



log = E snap

snap = dlc t sig


Sys
event
LS
Auditor
update?
log server
LS
Auditor
Genl
heckEvidence
sip
evidence
log = E snap
snap = dlc t sig



Sys

event

Auditor

update?

LS

log server

LS

snap$_A$

Auditor

Genl

heckEvidence

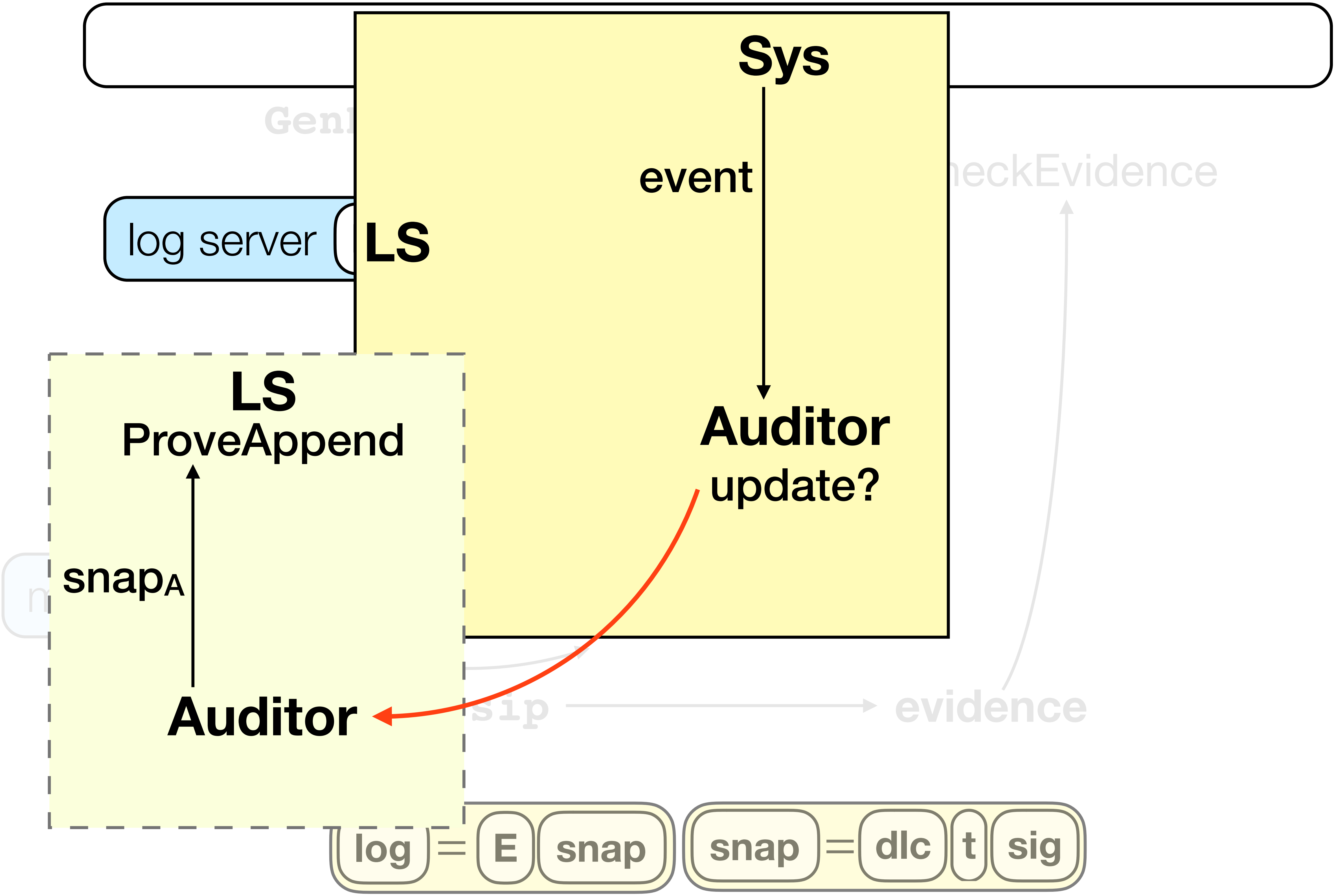
m

sip

evidence

log = E snap

snap = dlc t sig



Sys

event

Auditor

update?

LS

log server

LS

ProveAppend

snapA

Auditor

Genl

heckEvidence

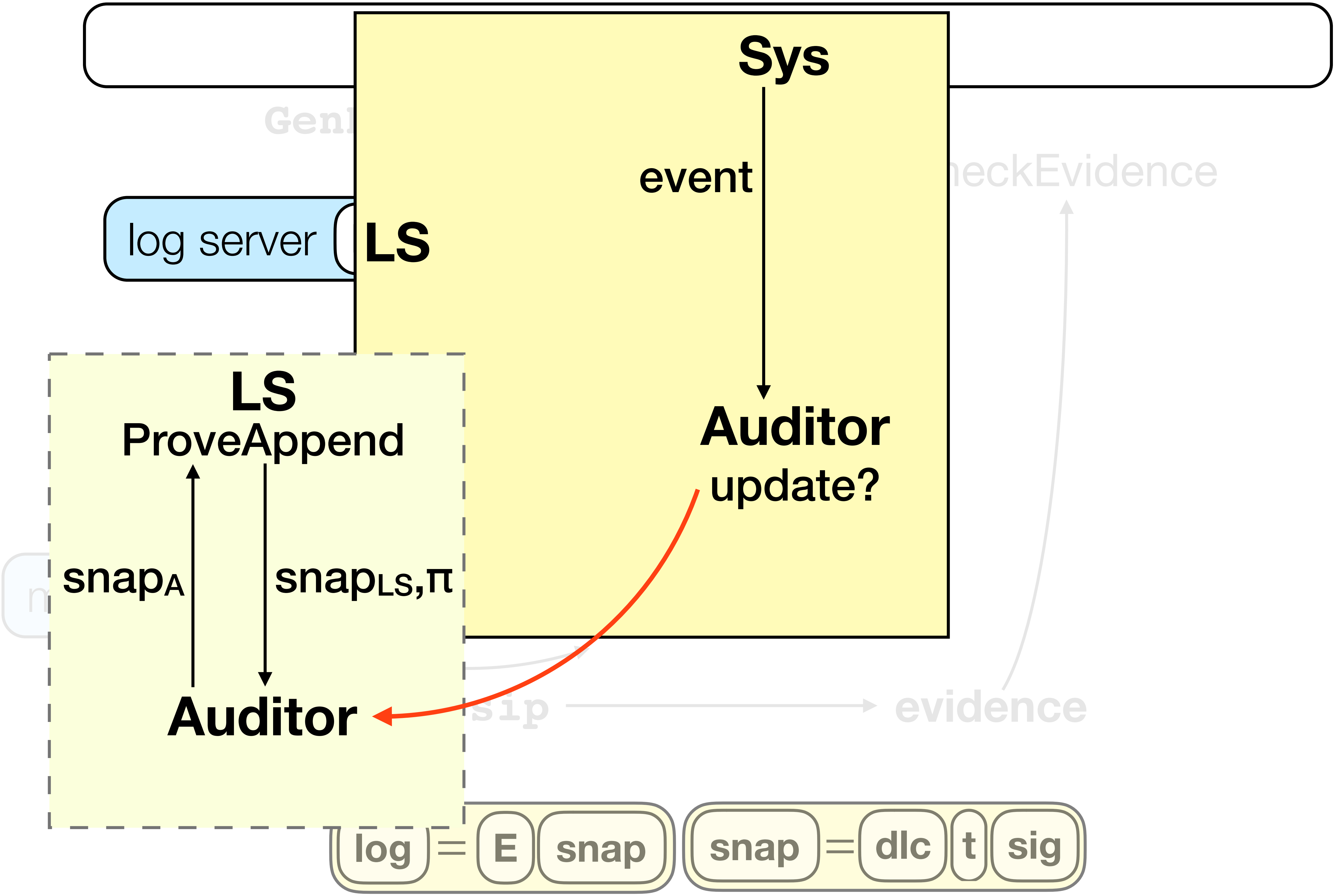
sip

evidence

n

log = E snap

snap = dlc t sig



Sys

event

Auditor

update?

log server

LS

Genl

heckEvidence

LS

ProveAppend

snap$_A$

snap$_{LS}$,π

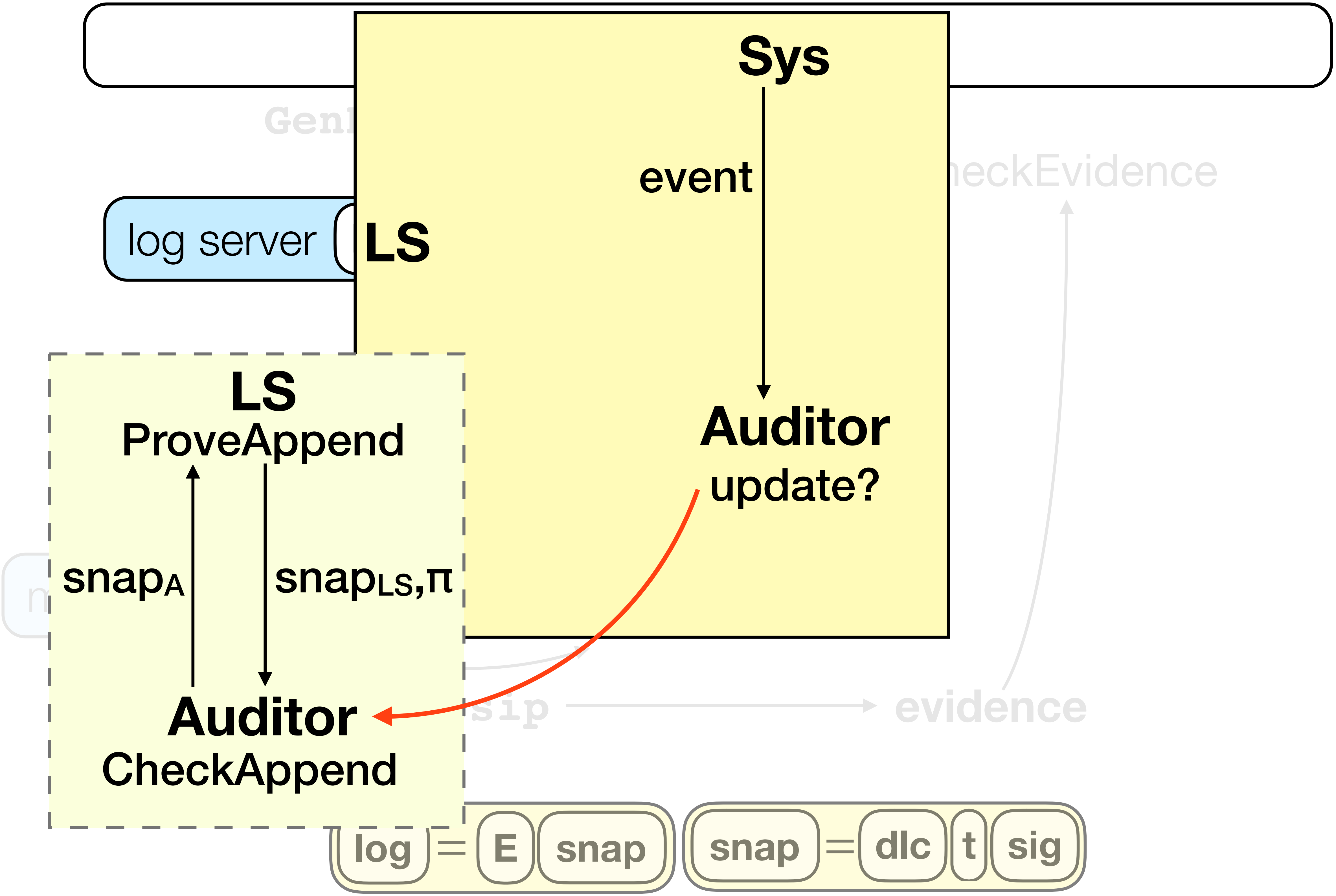
Auditor

sip

evidence

log = E snap

snap = dlc t sig



Sys

event

Auditor

update?

Gen

heckEvidence

log server

LS

LS

ProveAppend

snap$_A$

snap$_{LS}$,π

Auditor

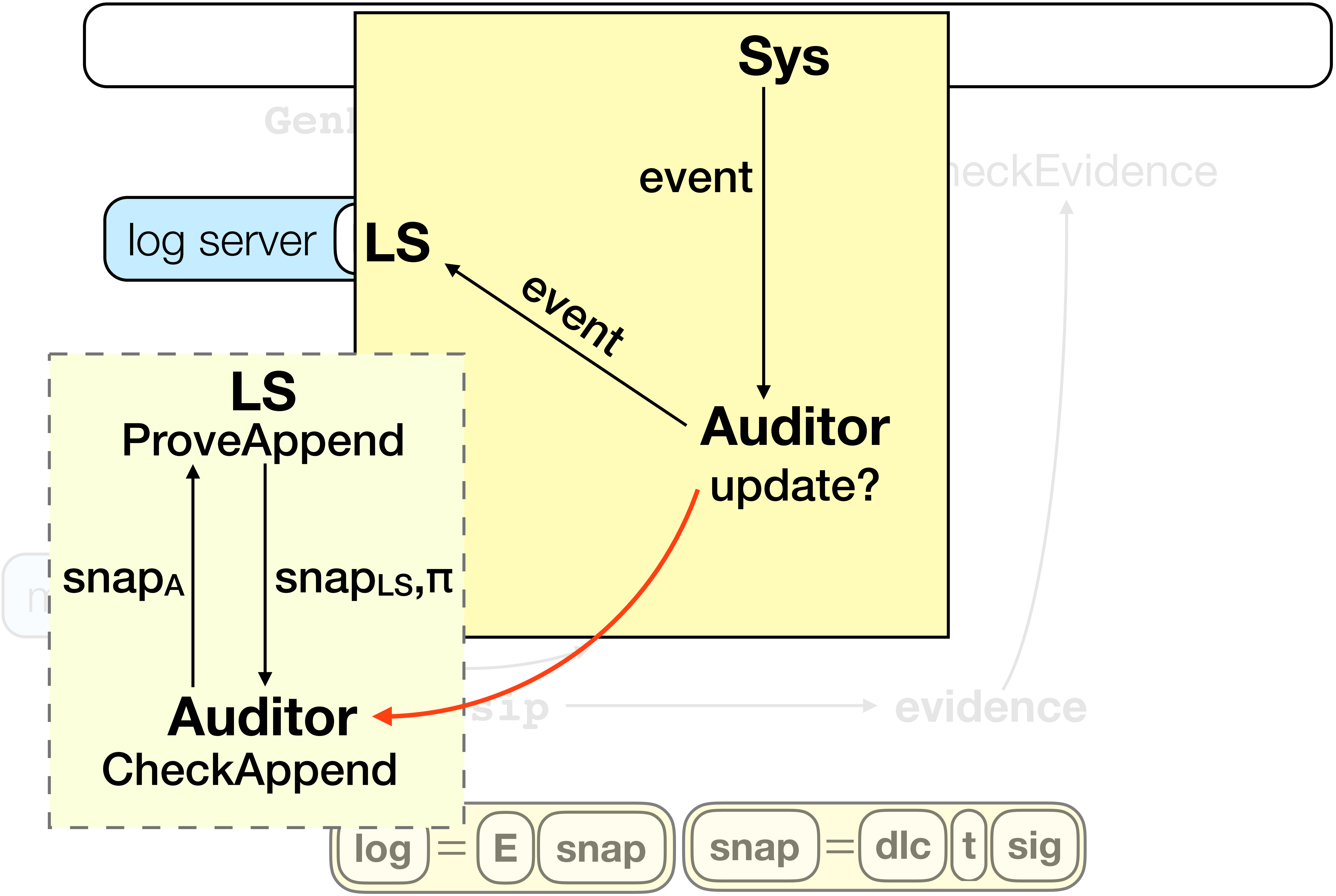
CheckAppend

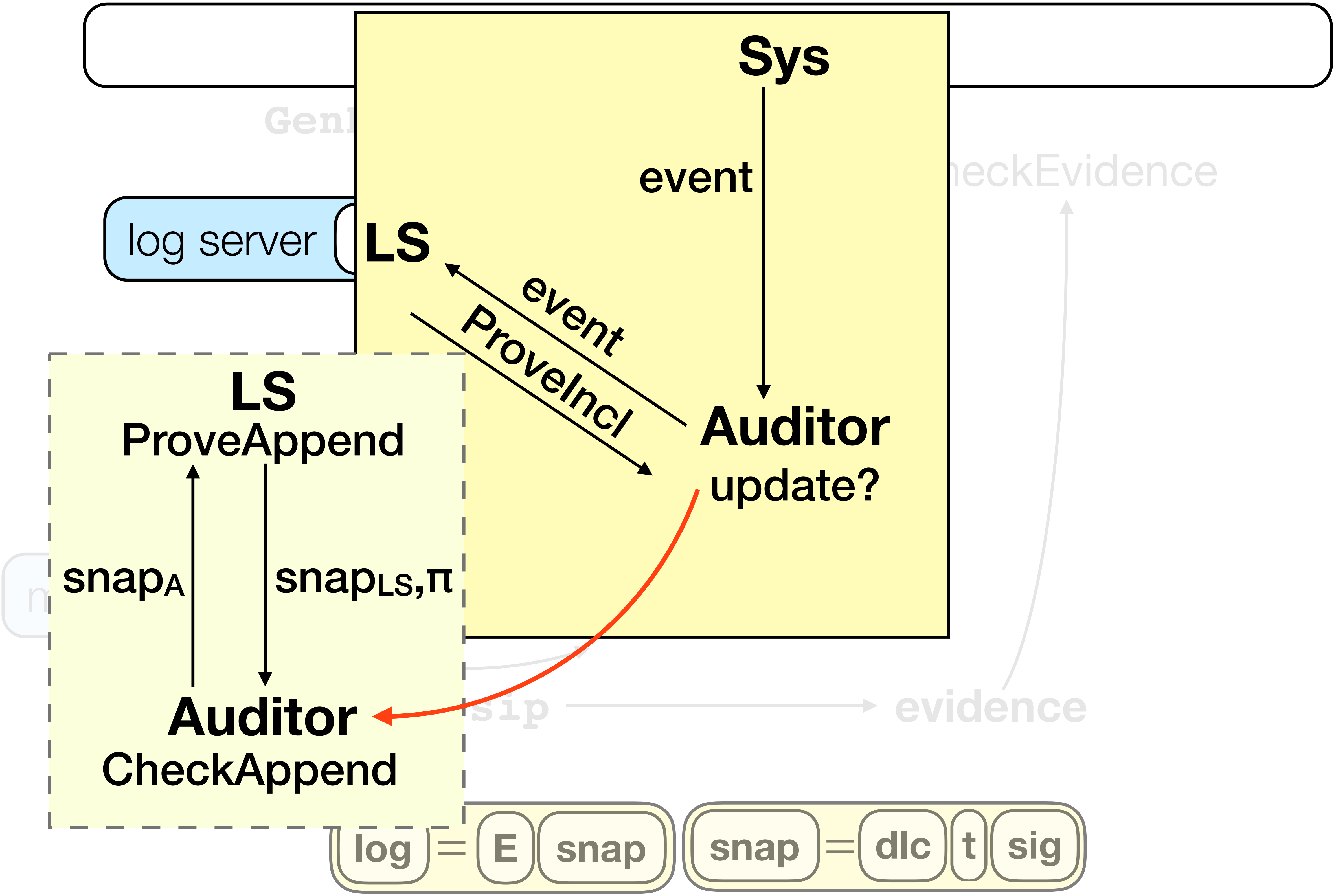
sip

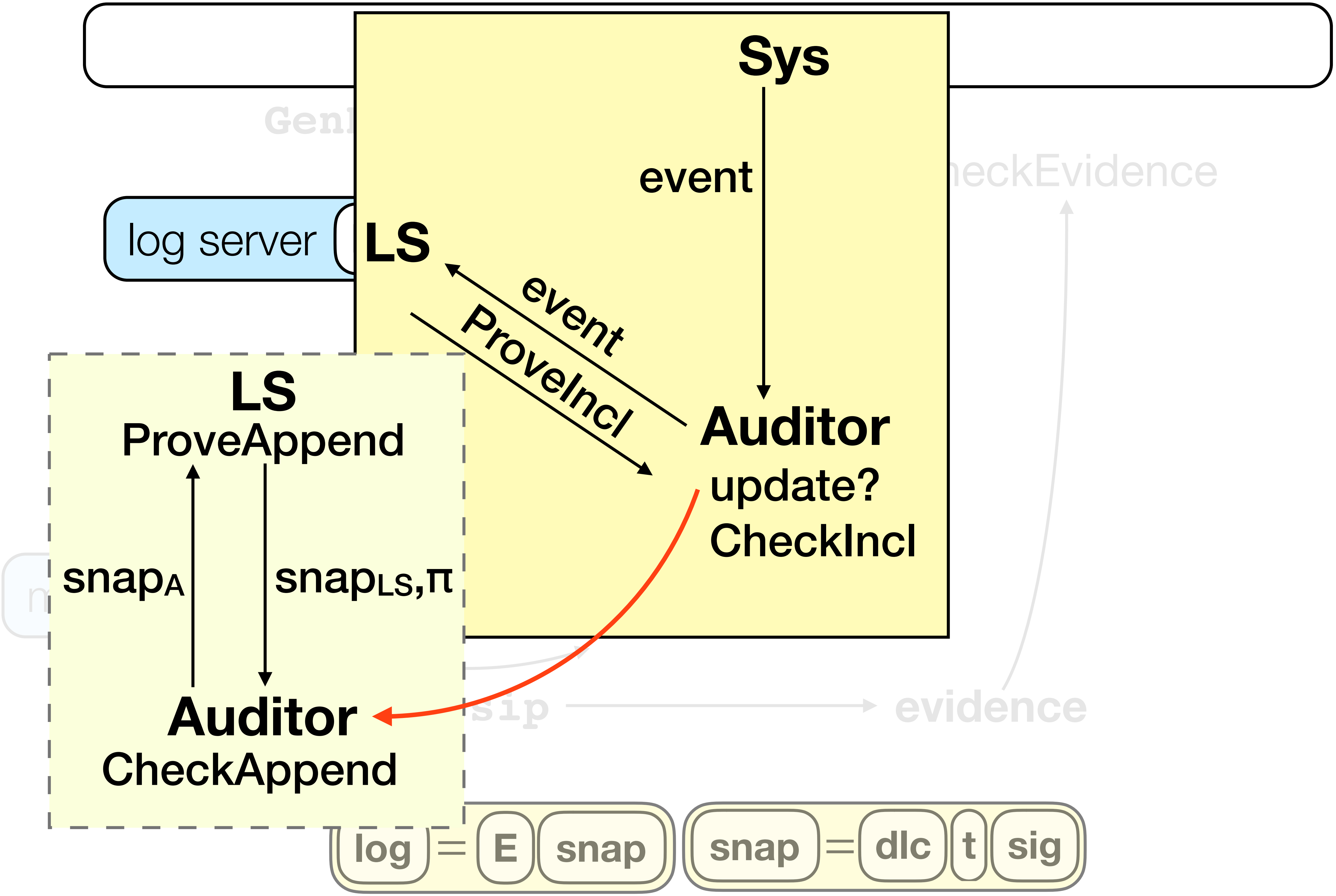
evidence

log = E snap

snap = dlc t sig


Sys
event
LS
event
Auditor
update?
LS
ProveAppend
snapA
snapLS,π
Auditor
CheckAppend
log server
GenI
heckEvidence
sip
evidence
log
=
E
snap
snap
=
dlc
t
sig


Sys
event
LS
event
ProveIncl
Auditor
update?
Genl
heckEvidence
log server
LS
ProveAppend
snapA
snapLS,π
Auditor
CheckAppend
sip
evidence
log = E snap
snap = dlc t sig



Sys

event

LS

event

ProveIncl

Auditor

update?

CheckIncl

log server

LS

ProveAppend

snap_A

snap_{LS},π

Auditor

CheckAppend

Genl

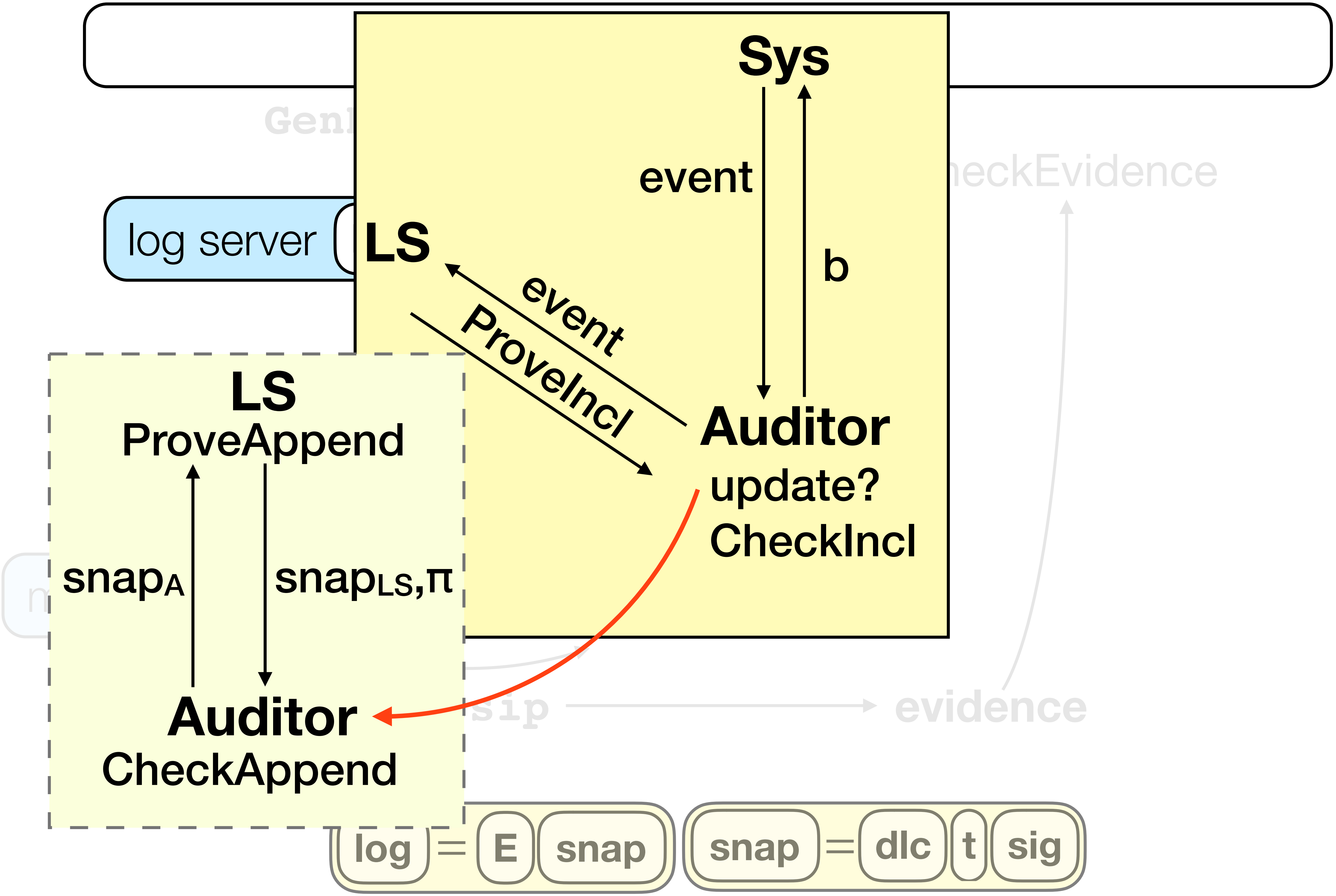
eckEvidence

sip

evidence

log = E snap

snap = dlc t sig



Sys

event

b

LS

event

ProveIncl

Auditor

update?

CheckIncl

log server

Gen

eckEvidence

LS

ProveAppend

snap_A

snap_{LS}, π

Auditor

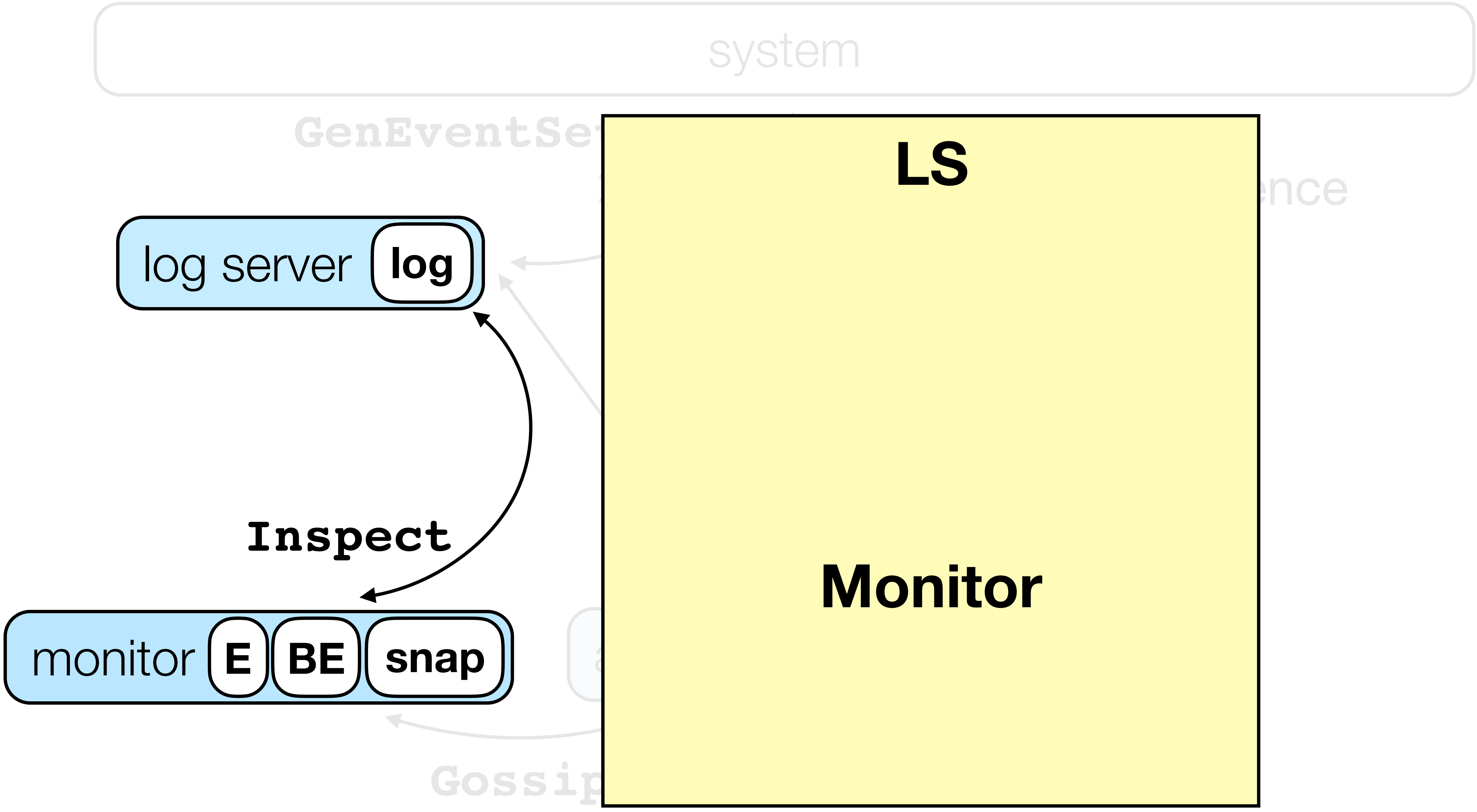
CheckAppend

sip

evidence

log = E snap

snap = dlc t sig


system

GenEventSe

ence

LS

Monitor

log server

log

Inspect

monitor

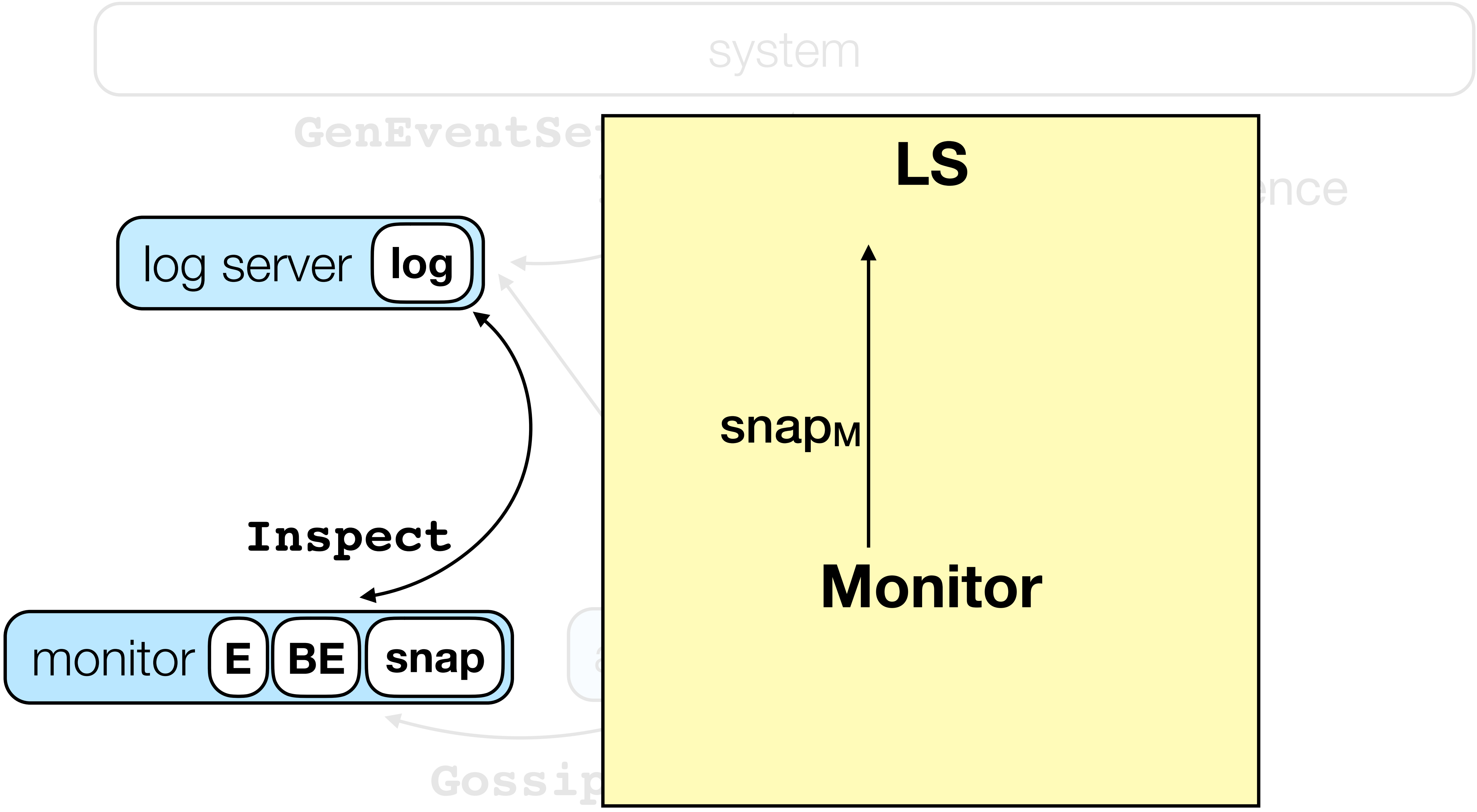
E

BE

snap

Gossip

log = E snap   snap = dlc t sig



system

GenEventSe

ence

log server log

Inspect

monitor E BE snap

a

Gossip

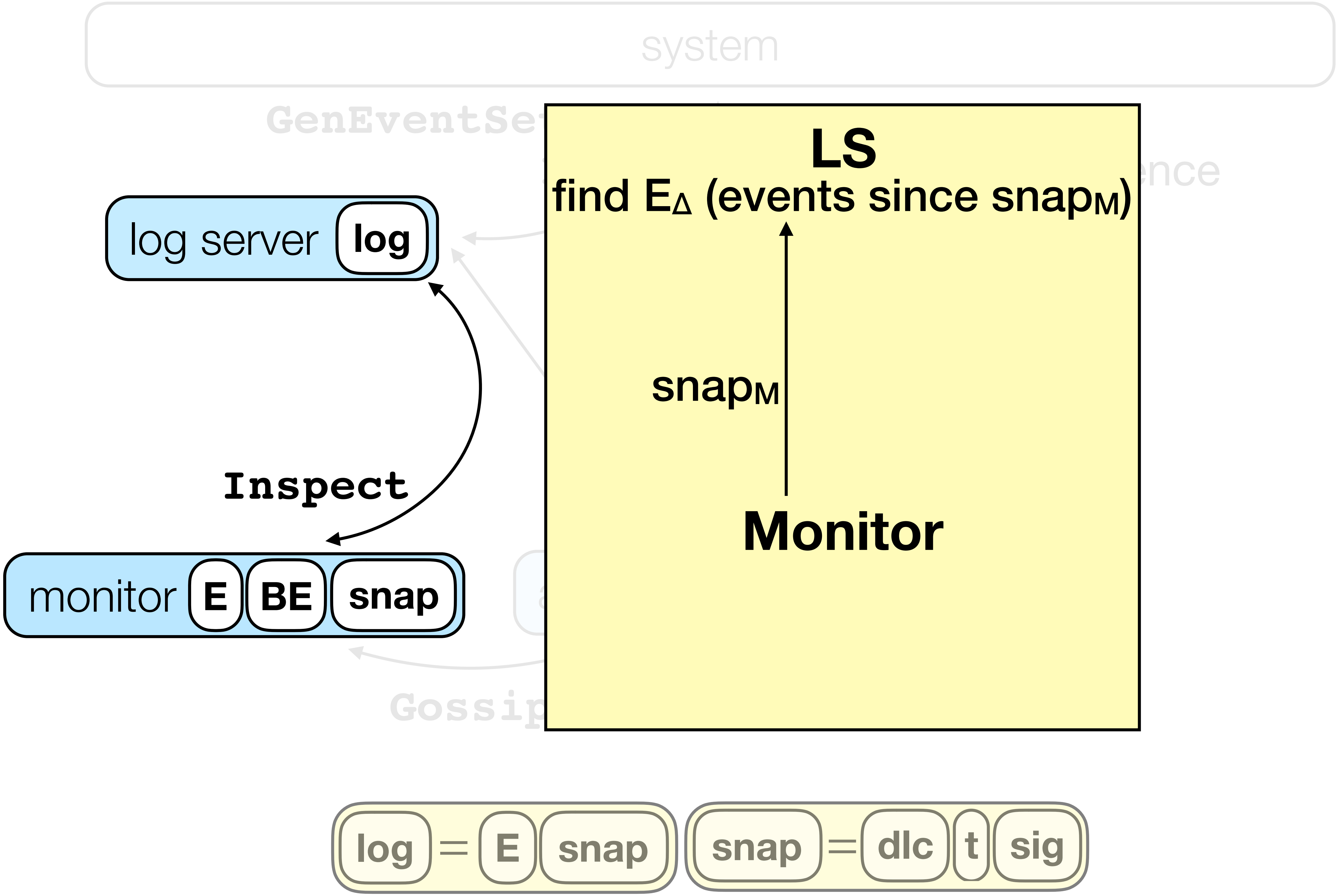
LS

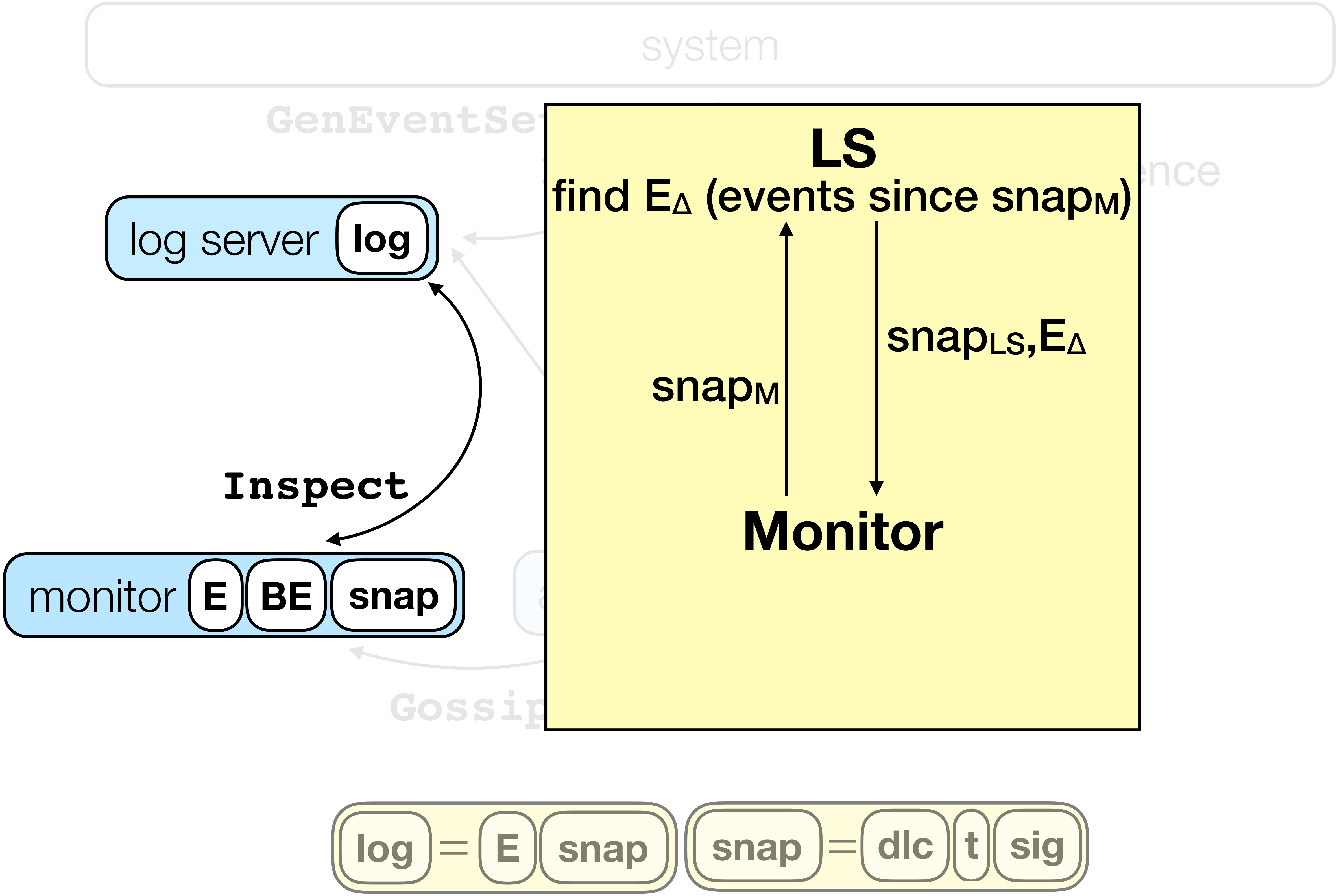
snap_M

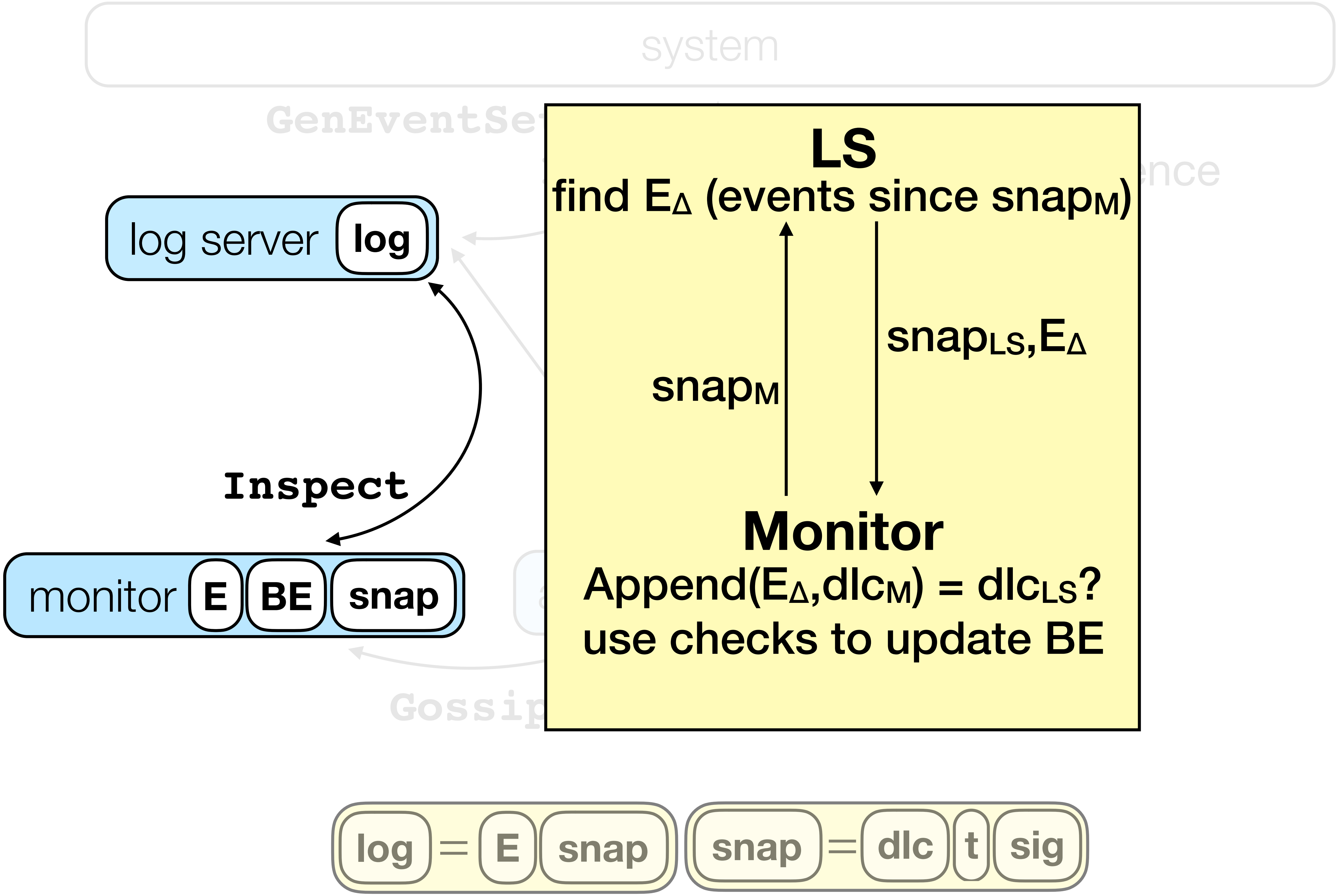
Monitor

log = E snap

snap = dlc t sig


system
GenEventSe
ence
log server
log
Inspect
monitor
E
BE
snap
Gossip
LS
find E_Δ (events since snap_M)
snap_M
Monitor
log
=
E
snap
snap
=
dlc
t
sig


system
GenEventSe
ence
log server
log
Inspect
monitor
E
BE
snap
Gossip
LS
find E_Δ (events since $snap_M$)
$snap_M$
$snap_{LS}, E_\Delta$
Monitor
log = E snap
snap = dlc t sig



system

GenEventSe

ence

log server log

Inspect

monitor E BE snap

Gossip

LS

find E_Δ (events since $snap_M$)

$snap_M$

$snap_{LS}$,E_Δ

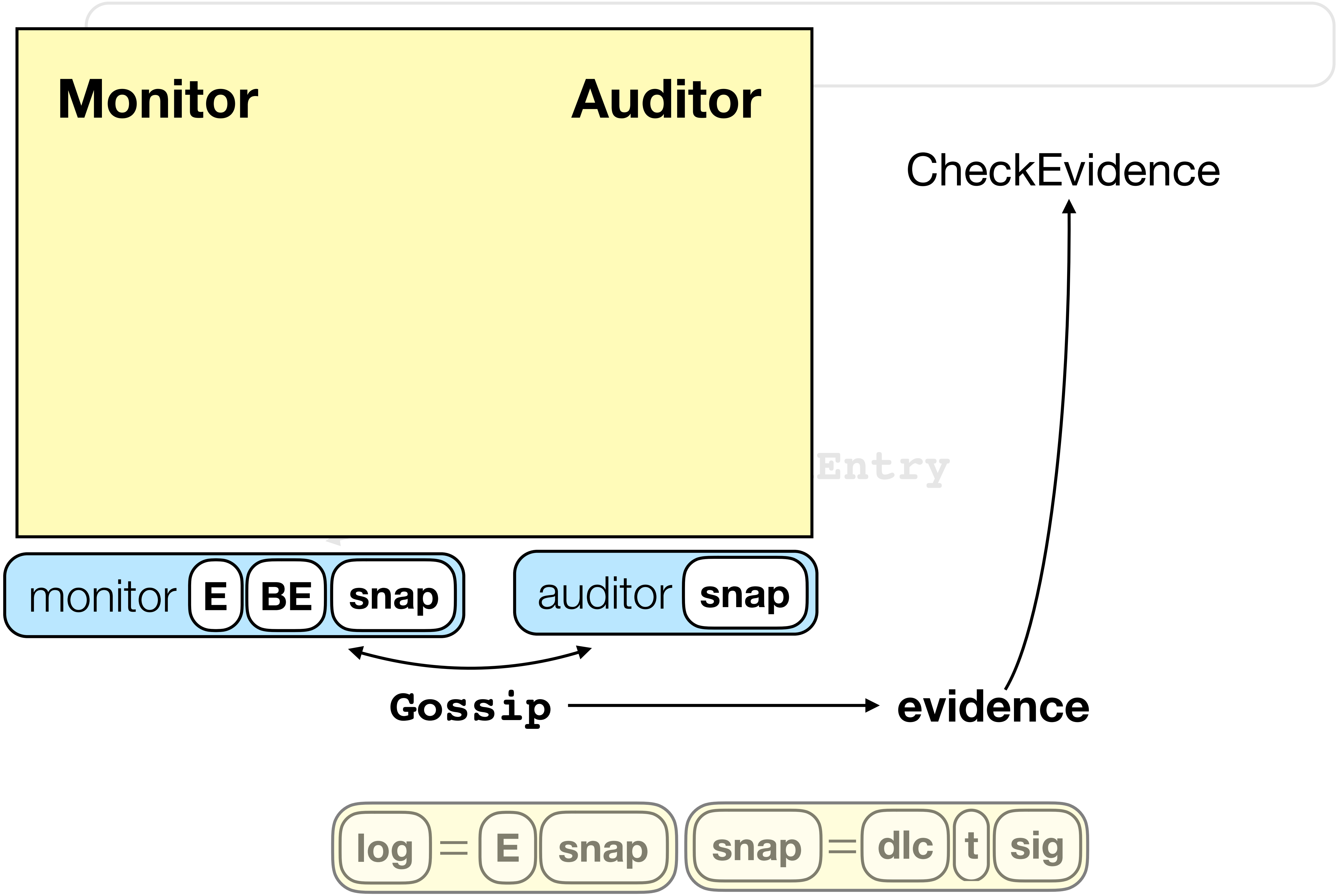
Monitor

Append(E_Δ,dlc_M) = dlc_{LS}?

use checks to update BE

log = E snap

snap = dlc t sig



Monitor

Auditor

CheckEvidence

Entry

monitor E BE snap

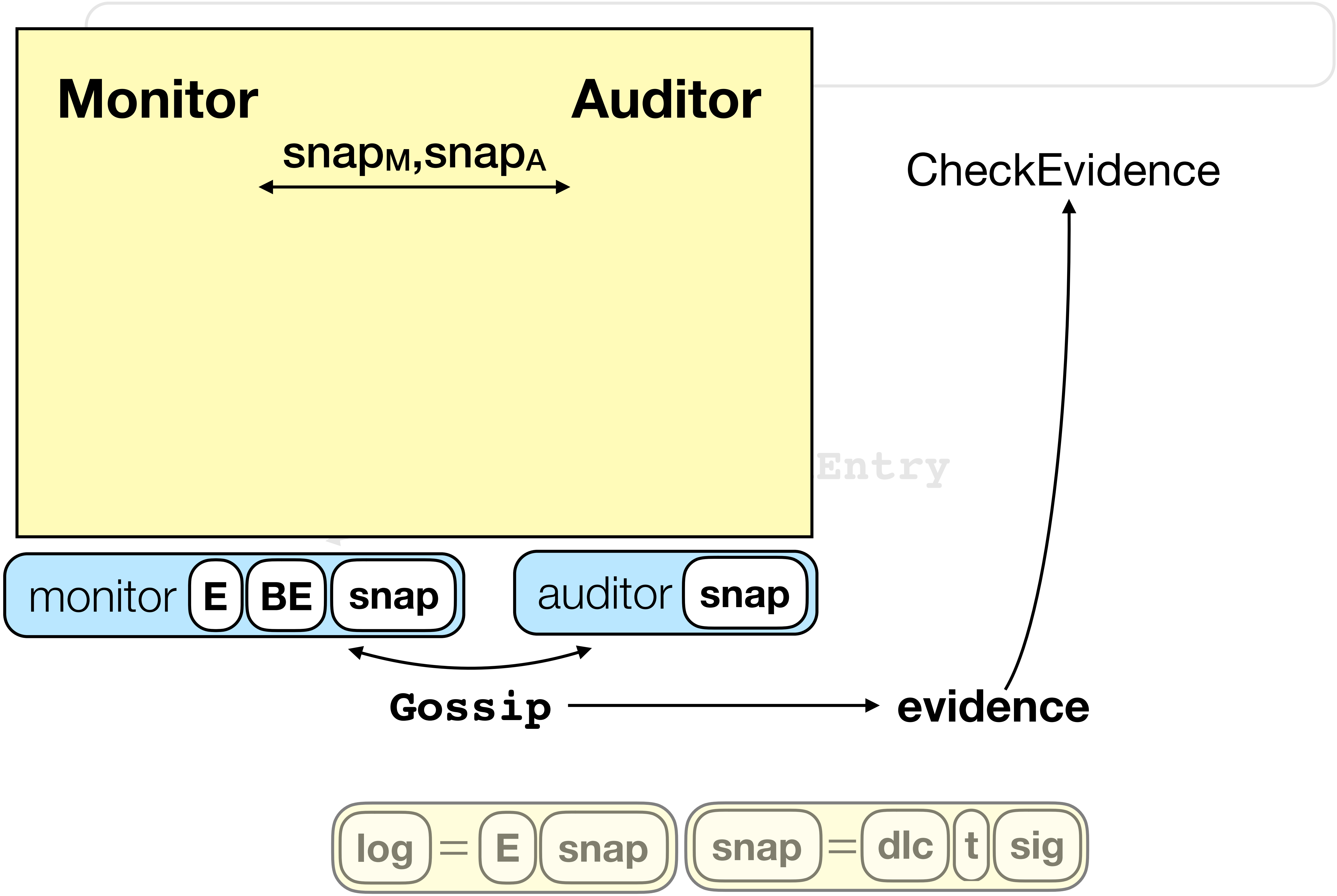
auditor snap

Gossip

evidence

log = E snap

snap = dlc t sig



Monitor

Auditor

$\text{snap}_M, \text{snap}_A$

CheckEvidence

Entry

monitor E BE snap

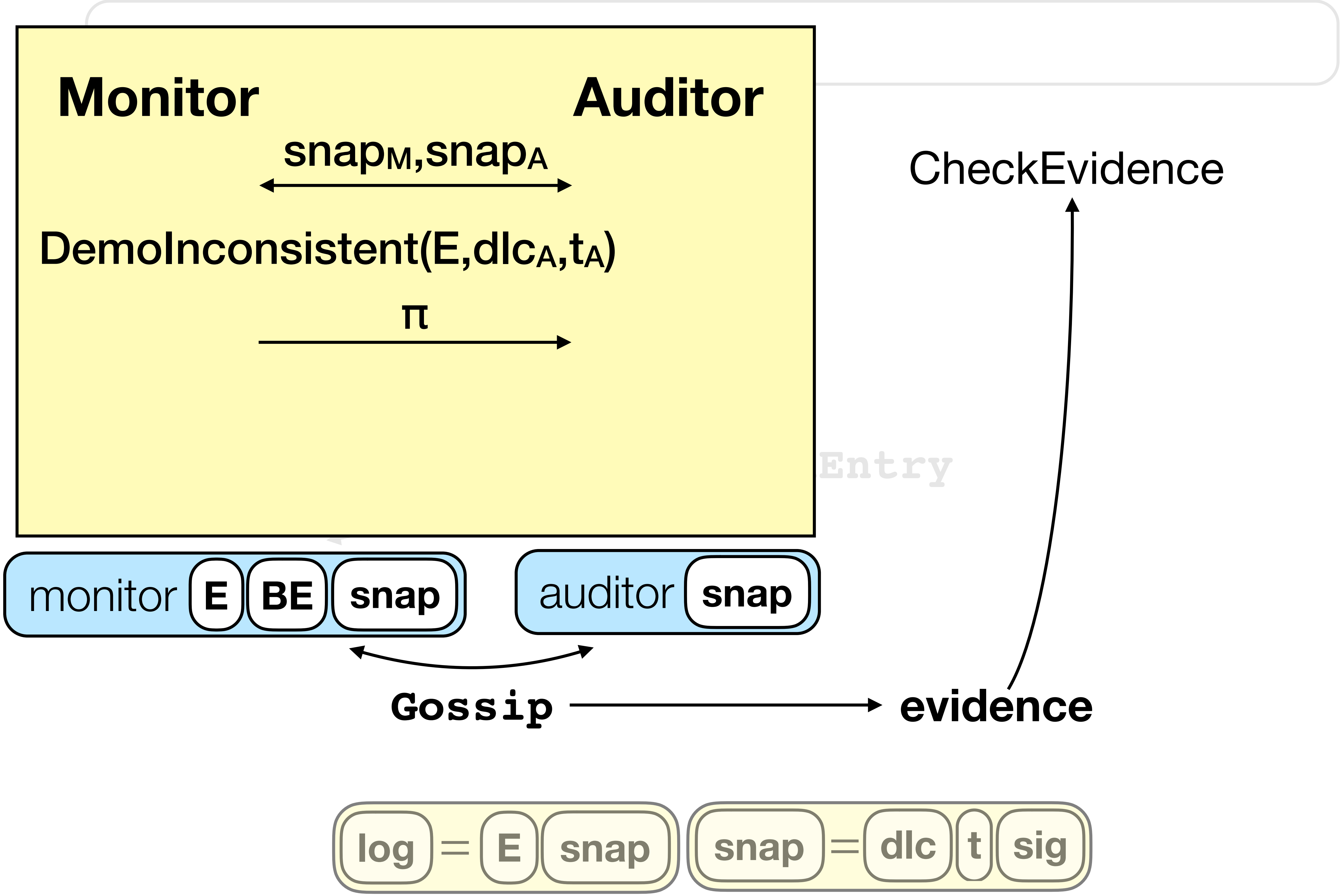
auditor snap

`Gossip`

evidence

log = E snap

snap = dlc t sig



Monitor **Auditor**

snap_M,snap_A

DemoInconsistent(E,dlc_A,t_A)

π

Entry

monitor **E** **BE** **snap**

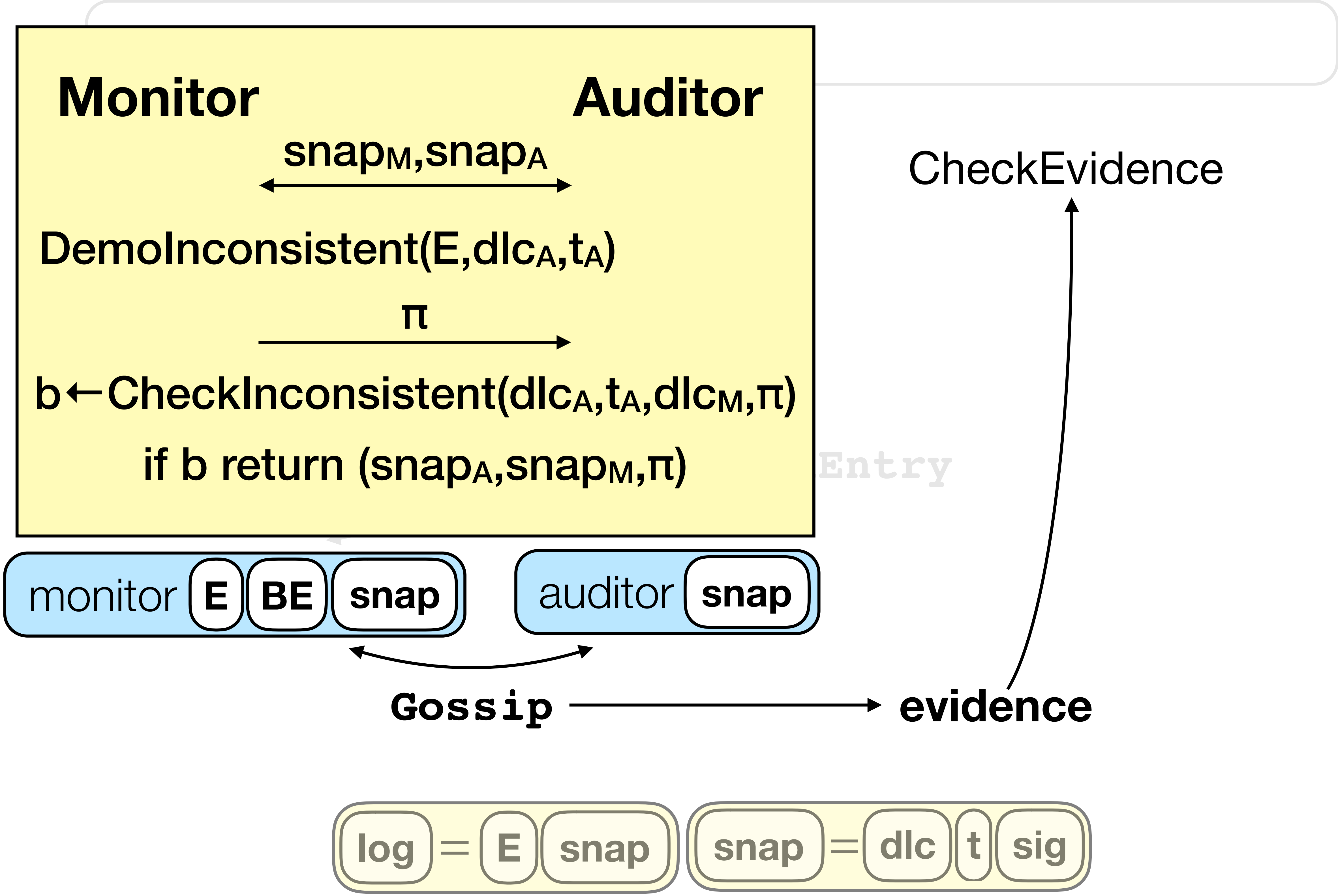
auditor **snap**

Gossip → **evidence**

CheckEvidence

log = E snap

snap = dlc t sig



Monitor **Auditor**

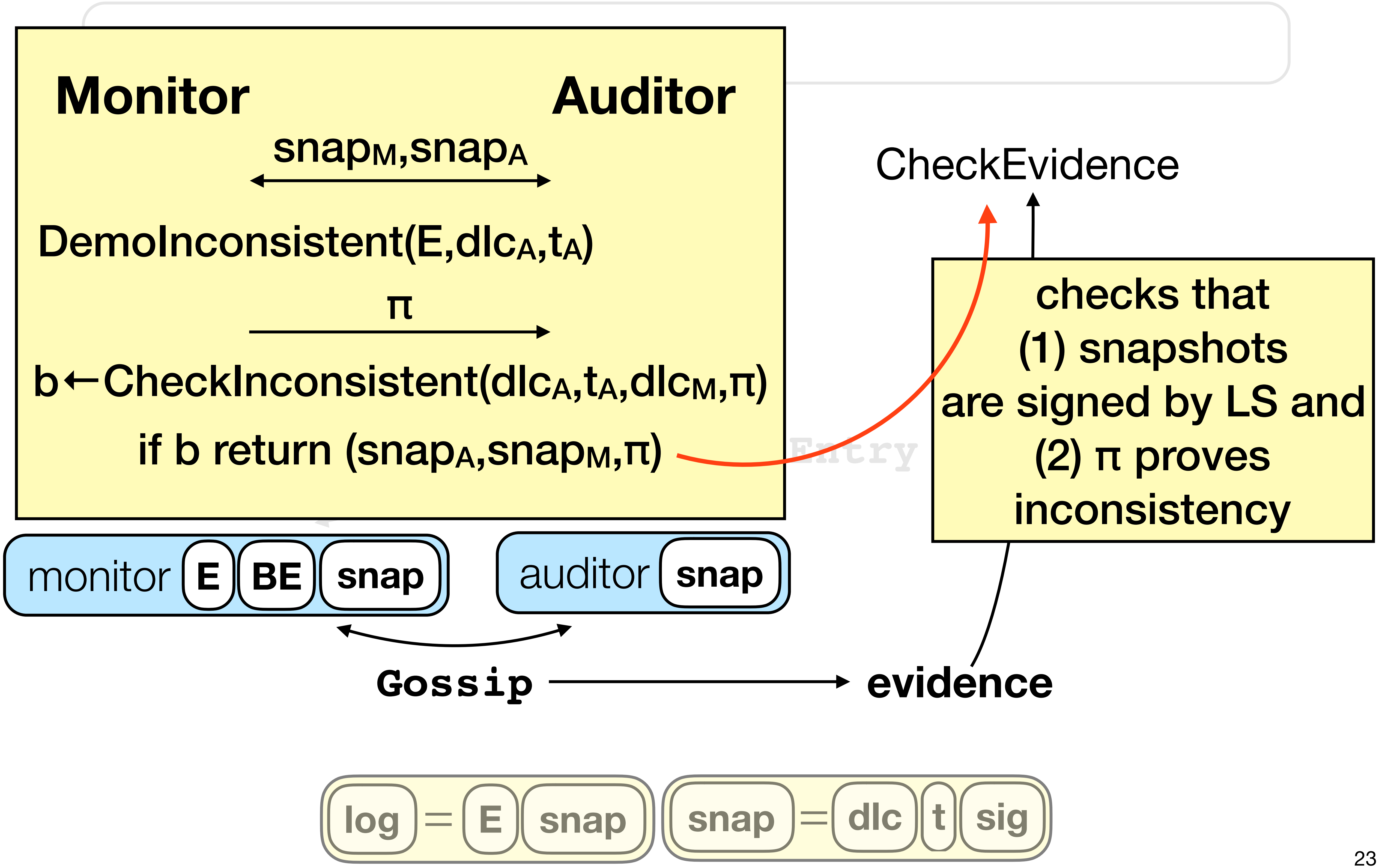
$\xleftrightarrow{snap_M, snap_A}$

DemoInconsistent(E, dlc_A, t_A)

$\xrightarrow{\pi}$

$b \leftarrow$ CheckInconsistent(dlc_A, t_A, dlc_M, π)

if b return ($snap_A, snap_M, \pi$)



Monitor **Auditor**

$\mathrm{snap}_M, \mathrm{snap}_A$

$\mathrm{DemoInconsistent}(E, \mathrm{dlc}_A, t_A)$

π

$b \leftarrow \mathrm{CheckInconsistent}(\mathrm{dlc}_A, t_A, \mathrm{dlc}_M, \pi)$

if b return $(\mathrm{snap}_A, \mathrm{snap}_M, \pi)$

CheckEvidence

checks that
(1) snapshots
are signed by LS and
(2) π proves
inconsistency

monitor E BE snap

auditor snap

Gossip → **evidence**

log = E snap

snap = dlc t sig

security

ability to carry out
DemoInconsistent, ProveAppend, and ProveIncl ⇒

consistency

unforgeability of
DemoInconsistent, DemoNotIncl*, and signature scheme ⇒

non-frameability

ability to carry out DemoNotIncl* ⇒

accountability

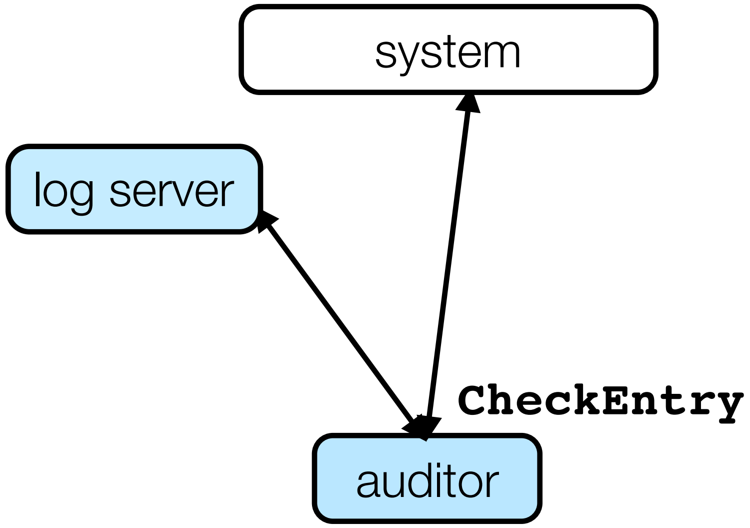
*uses pledged version in which Auditor keeps track of failed events and gossips about them with Monitor to produce new type of evidence

goal: bad events are exposed

system receives promises to include events in the log

goal: bad events are exposed

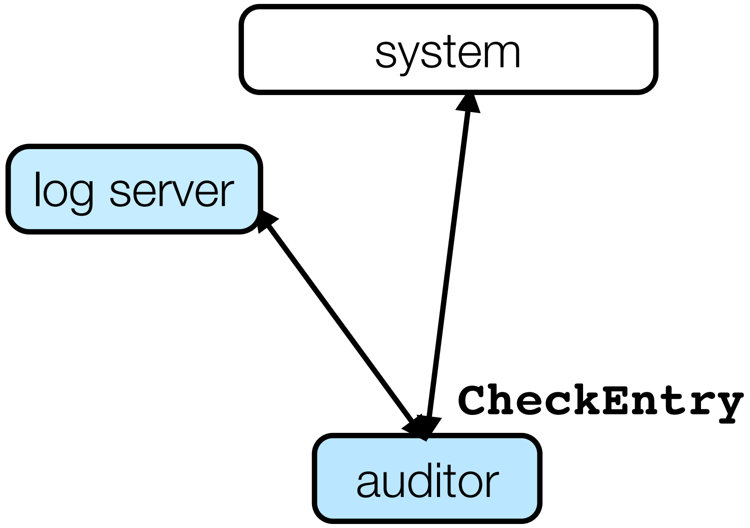
system receives promises to include events in the log



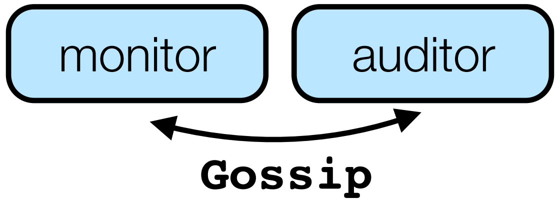
+ auditors determine if these events are in the log

goal: bad events are exposed

system receives promises to include events in the log



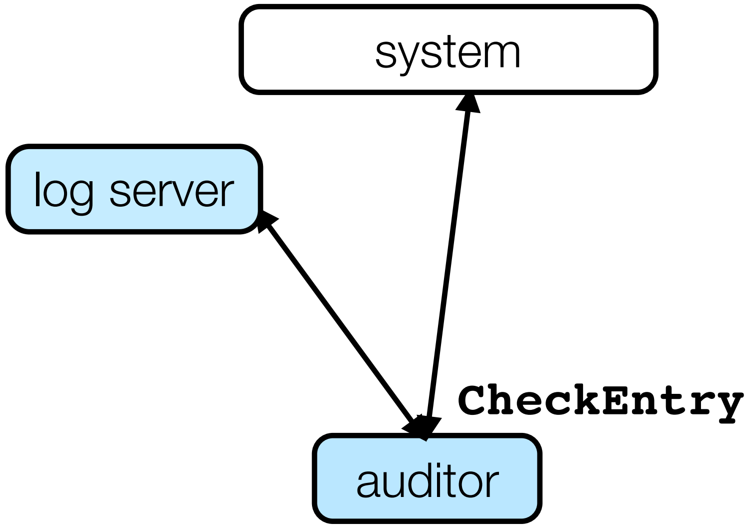
+ auditors determine if these events are in the log



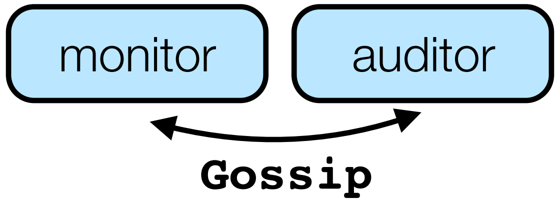
+ auditors and monitors ensure consistent view of log

goal: bad events are exposed

system receives promises to include events in the log



+ auditors determine if these events are in the log



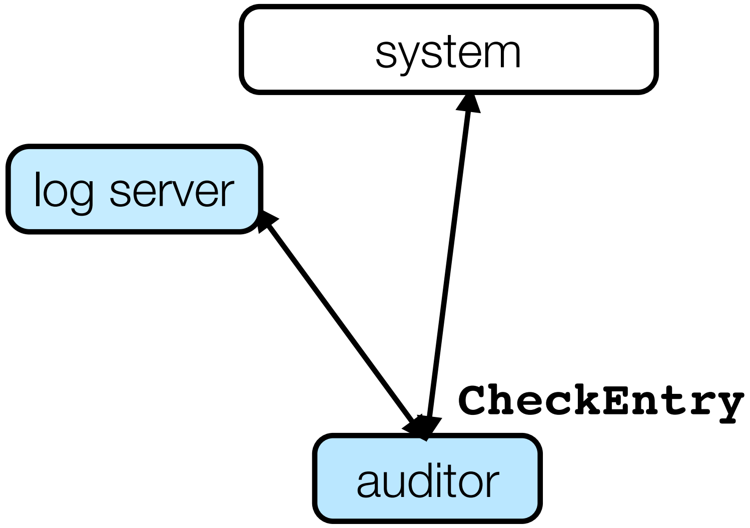
+ auditors and monitors ensure consistent view of log

⇒ (by consistency+accountability)

event is in monitor’s view of the log

goal: bad events are exposed

system receives promises to include events in the log



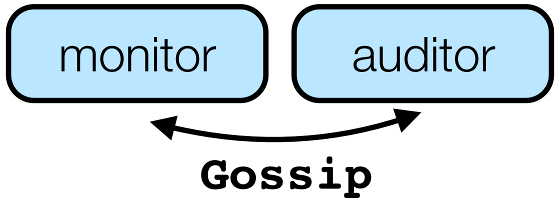
system

log server

CheckEntry

auditor

+ auditors determine if these events are in the log



monitor

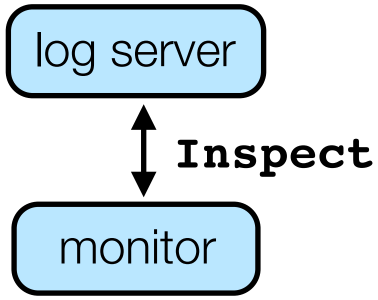
auditor

Gossip

+ auditors and monitors ensure consistent view of log

⇒ (by consistency+accountability)

event is in monitor’s view of the log



log server

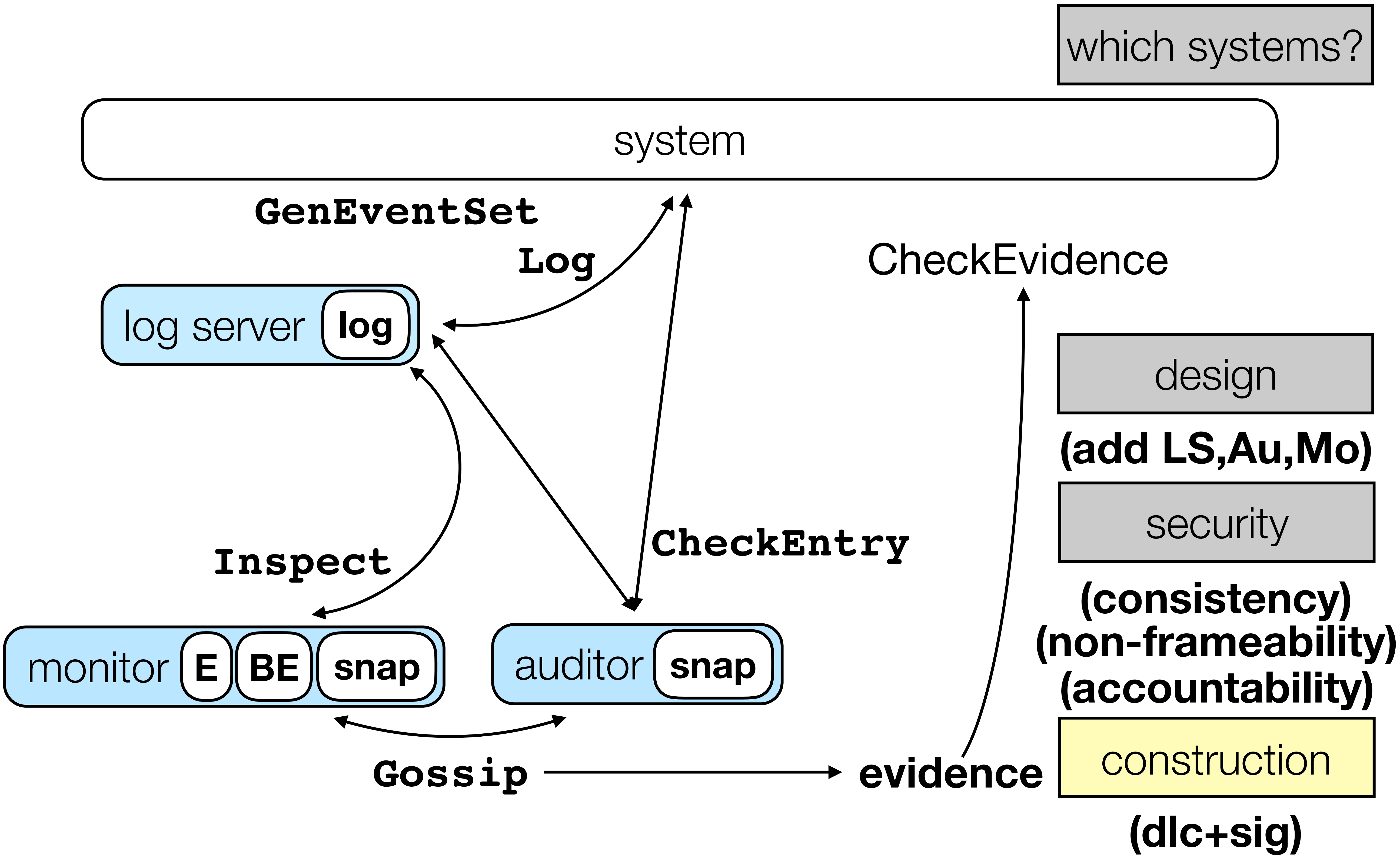
Inspect

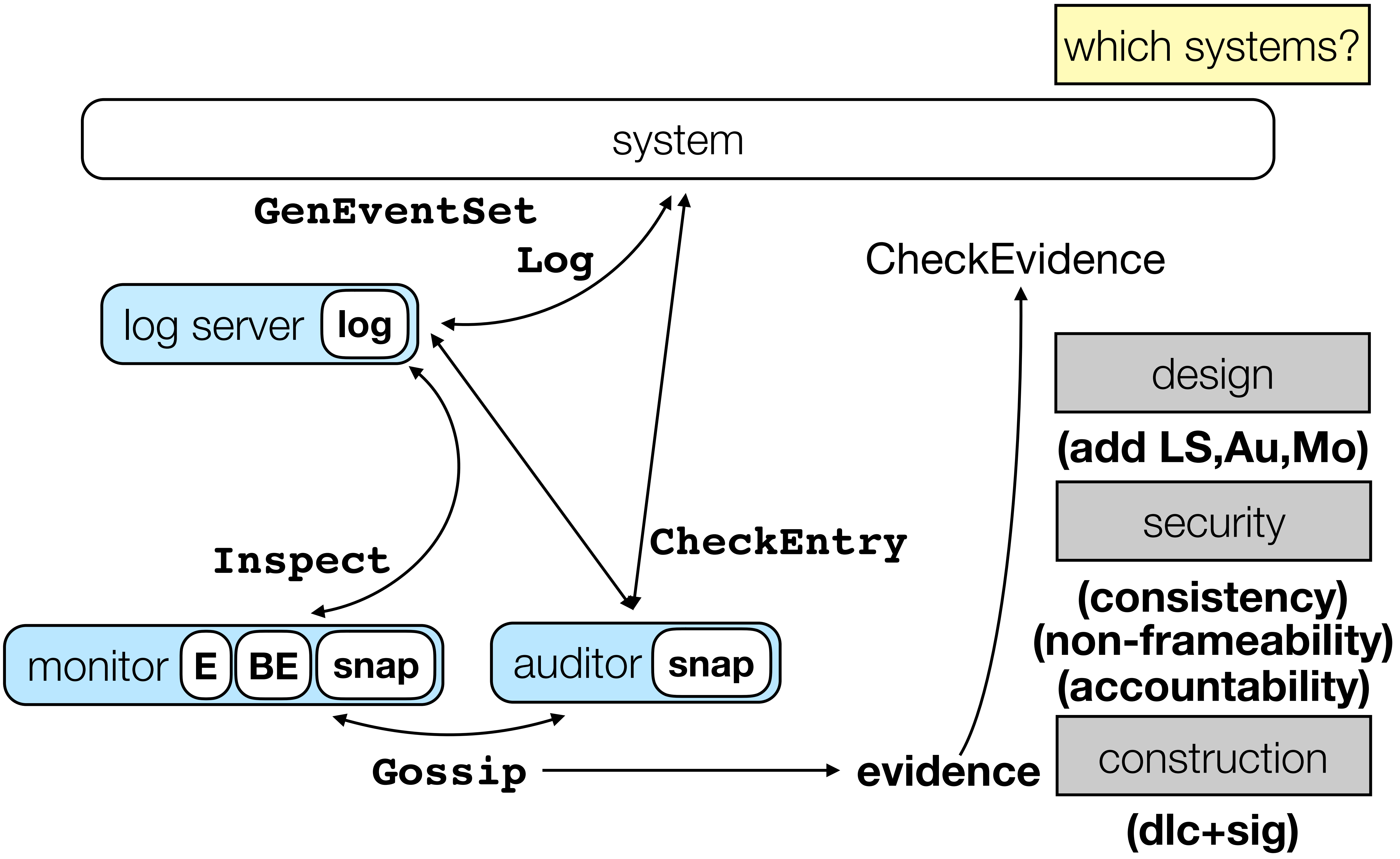
monitor

+ monitors detect bad events in the log

⇒

goal: bad events are exposed


which systems?
system
GenEventSet
Log
CheckEvidence
log server
log
design
(add LS,Au,Mo)
security
CheckEntry
Inspect
(consistency)
(non-frameability)
(accountability)
monitor
E
BE
snap
auditor
snap
construction
Gossip
evidence
(dlc+sig)



which systems?

system

GenEventSet

Log

CheckEvidence

log server log

design

(add LS,Au,Mo)

CheckEntry

Inspect

security

(consistency)
(non-frameability)
(accountability)

monitor E BE snap

auditor snap

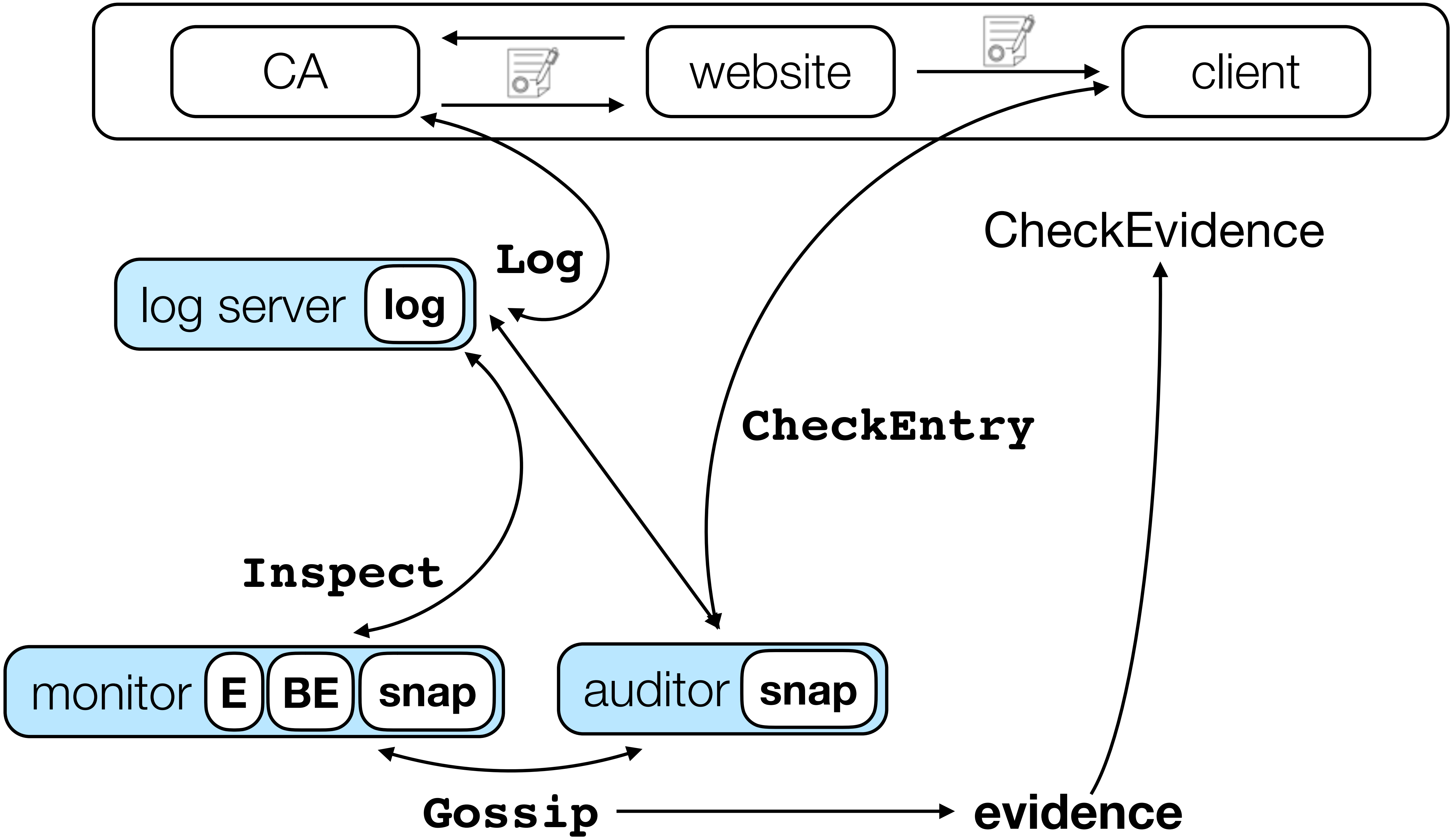
Gossip

evidence

construction

(dlc+sig)

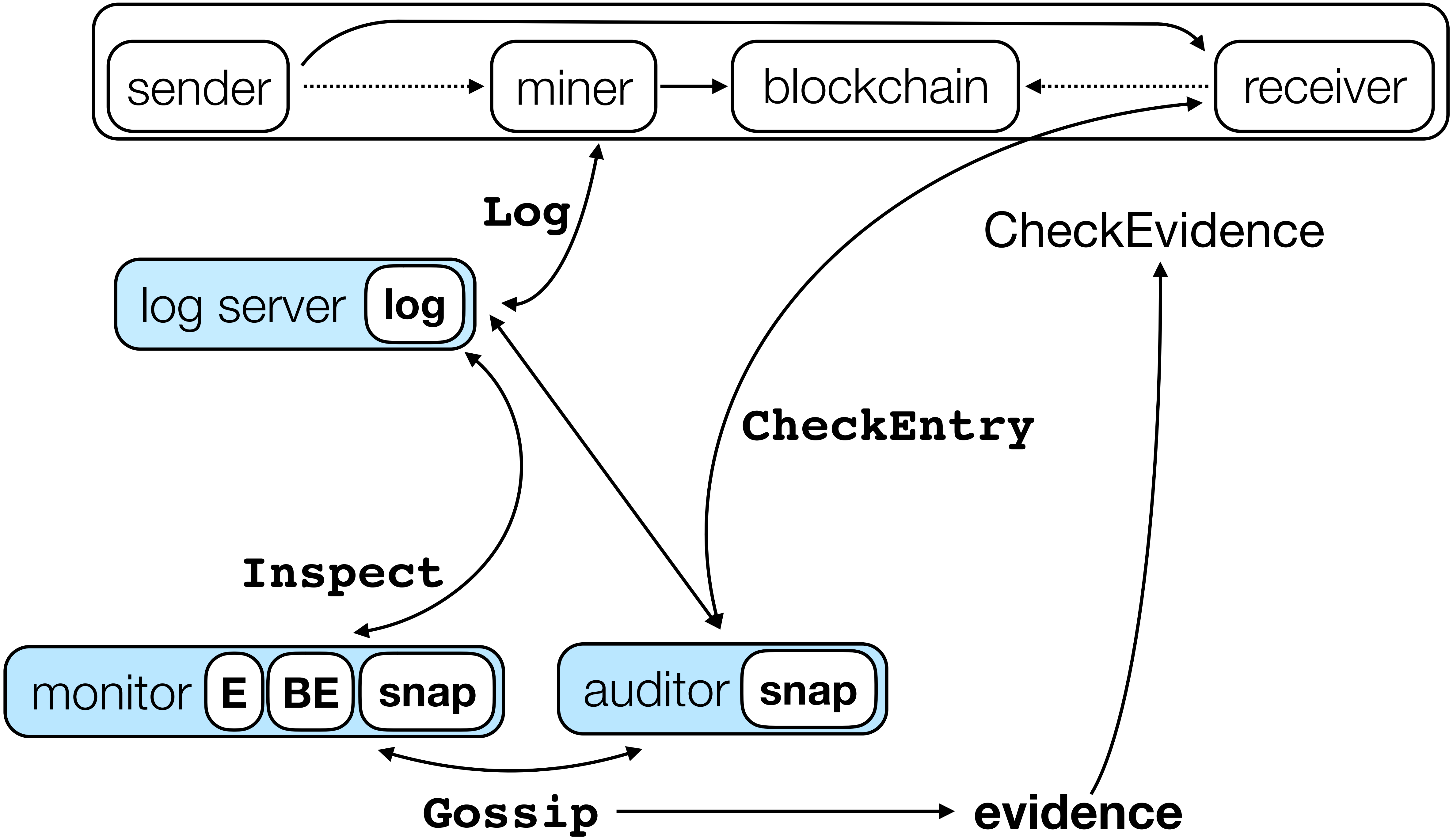
Certificate Transparency



bad certificate issuance is exposed
⇒ clients are less likely to accept bad certificates

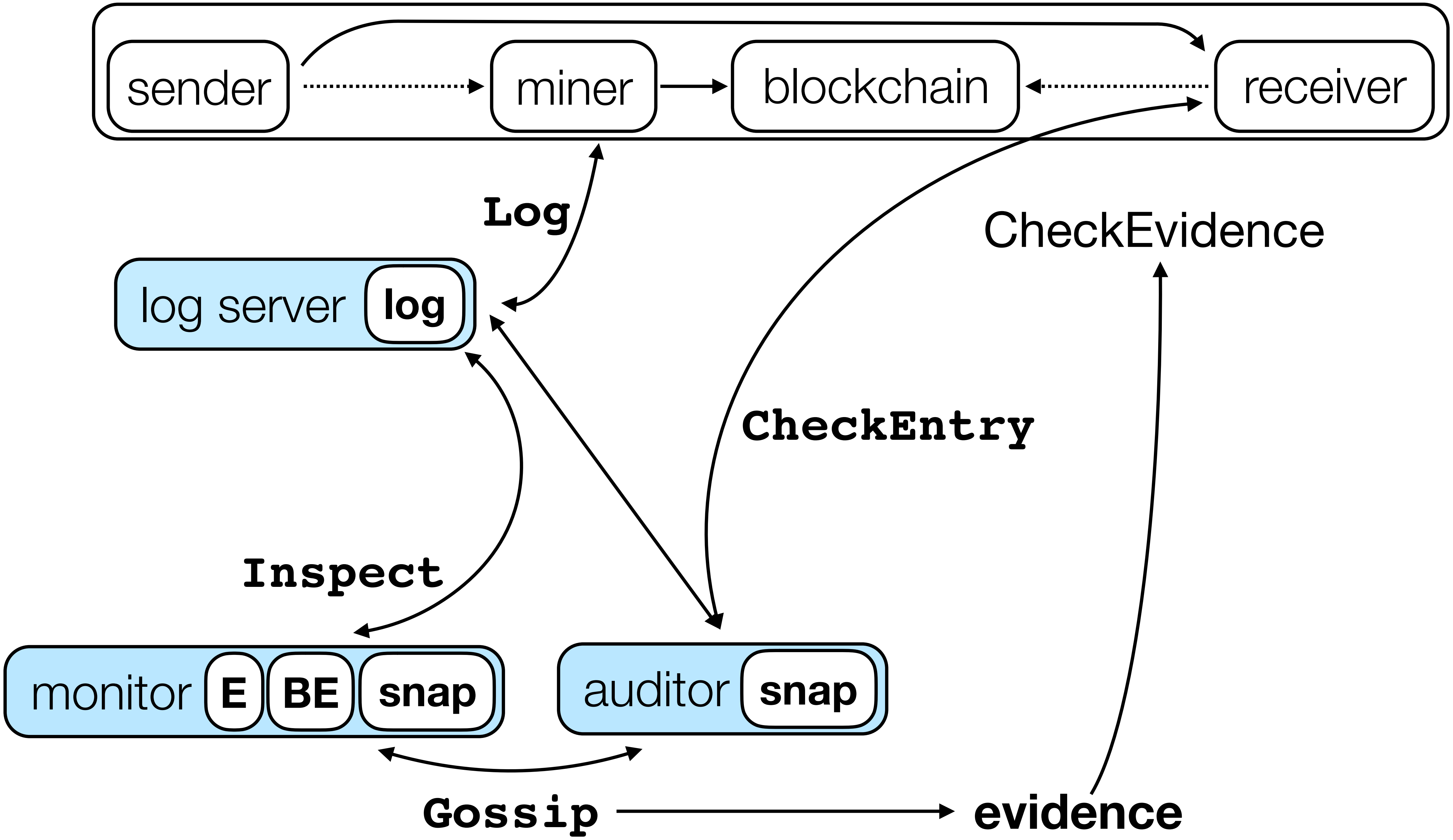
(icon by parkjisun from noun project)

Bitcoin



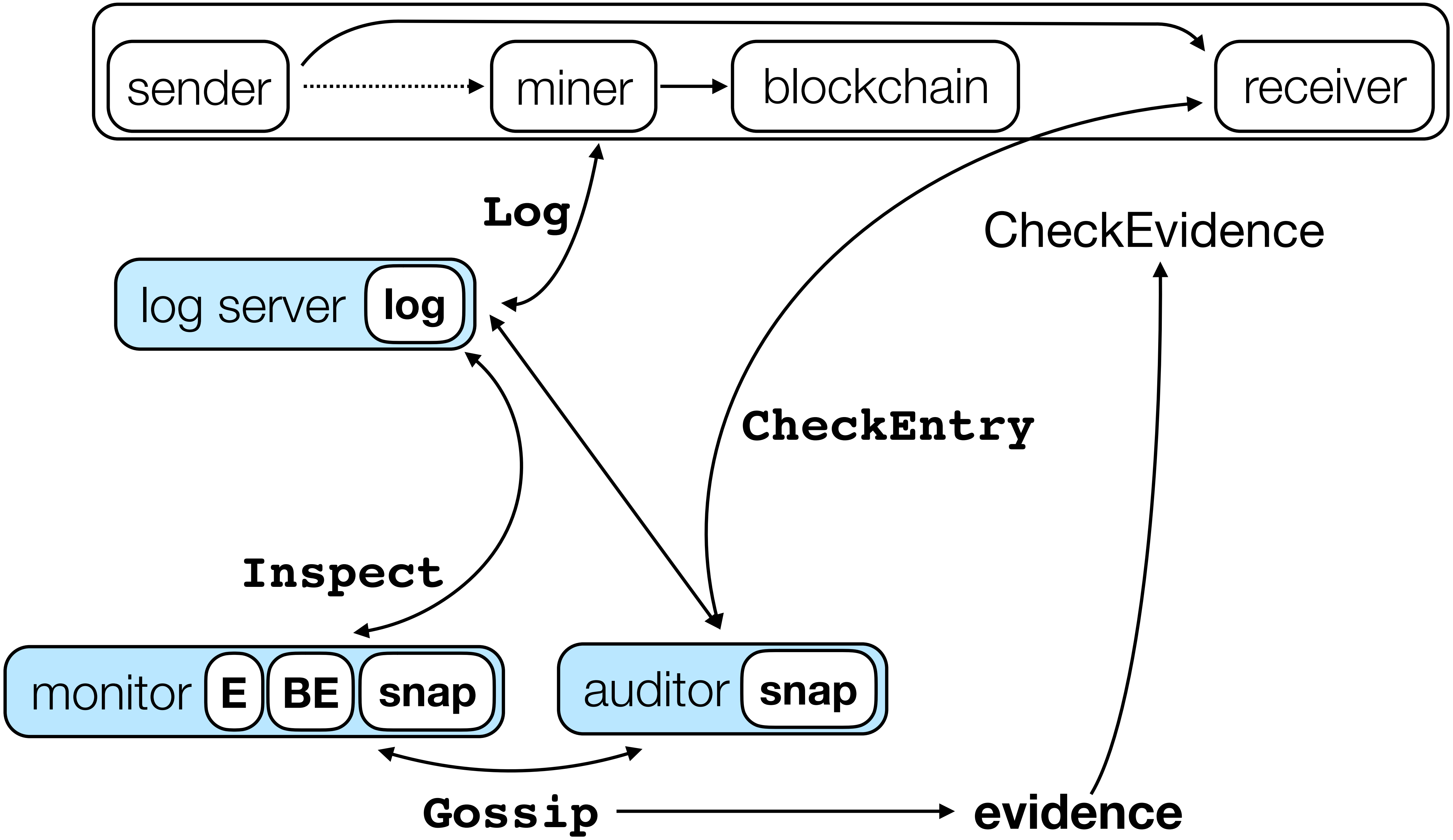
double spending is exposed

Bitcoin



double spending is exposed … provably!

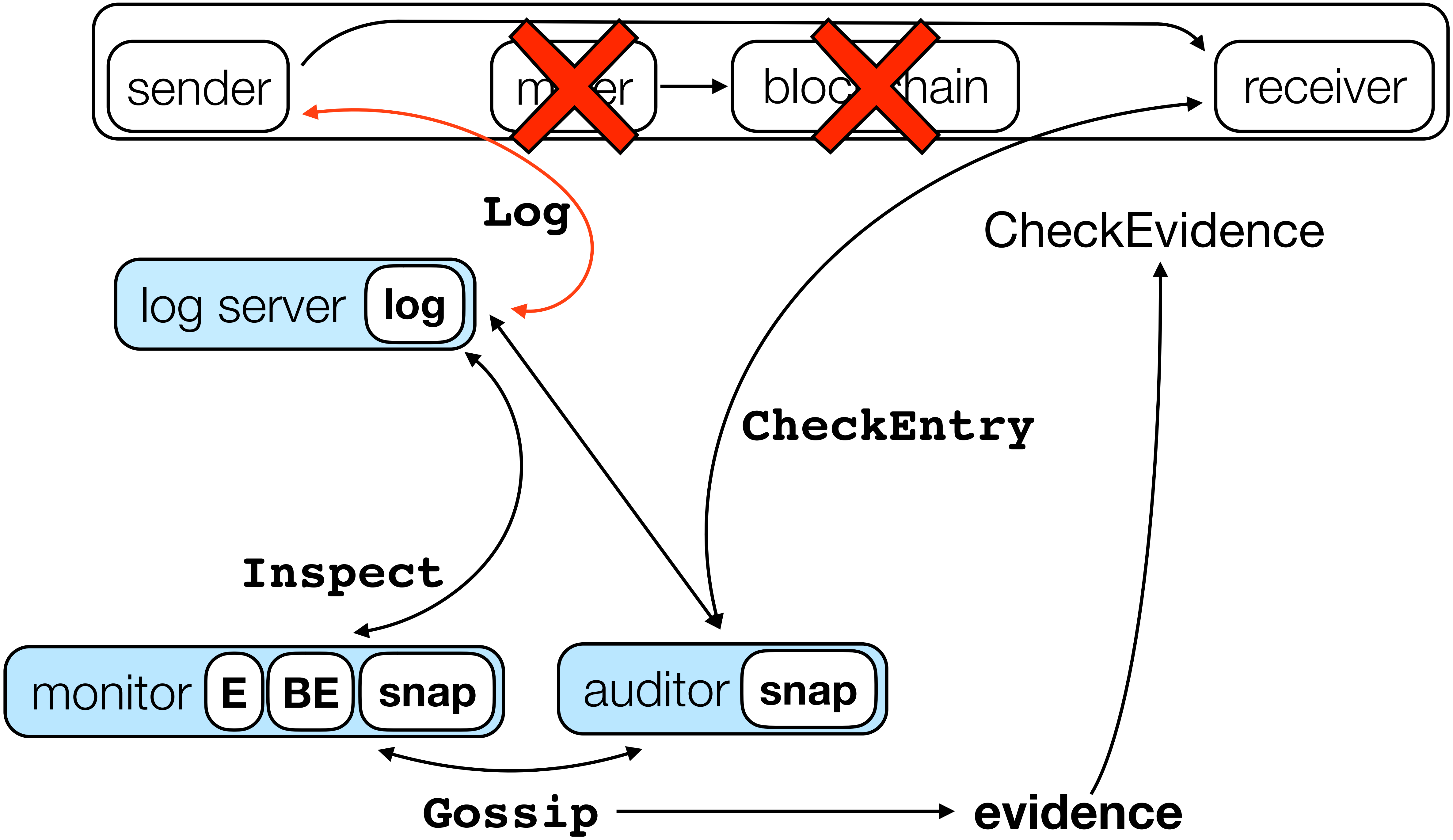
Bitcoin



double spending is exposed … provably!

sender and receiver don’t need to store blockchain

Bitcoin



double spending is exposed … provably!

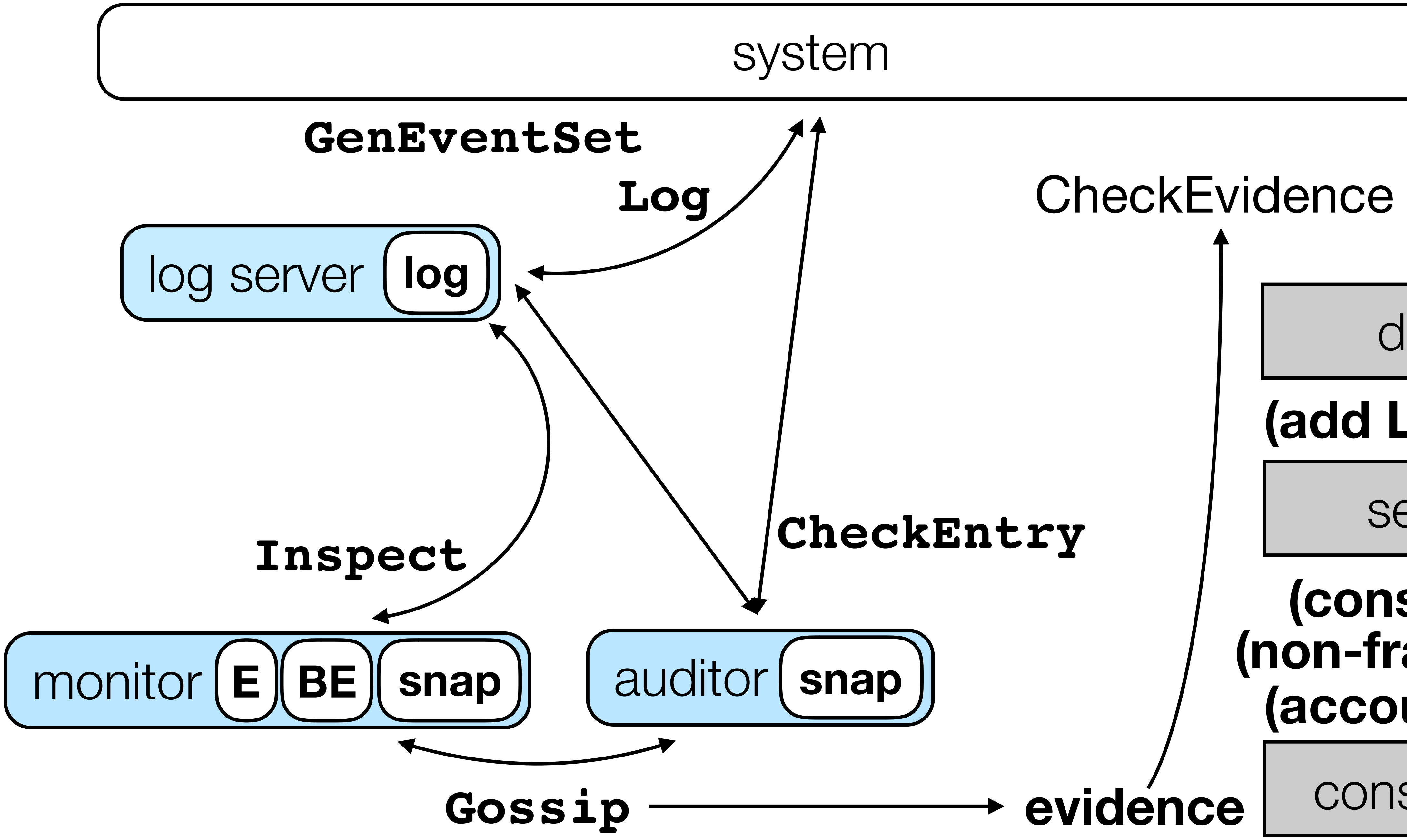
sender and receiver don’t need to store blockchain

gives rise to hybrid system with no mining

open problems

(CT+Bitcoin)

which systems?



design

(add LS,Au,Mo)

security

(consistency)
(non-frameability)
(accountability)

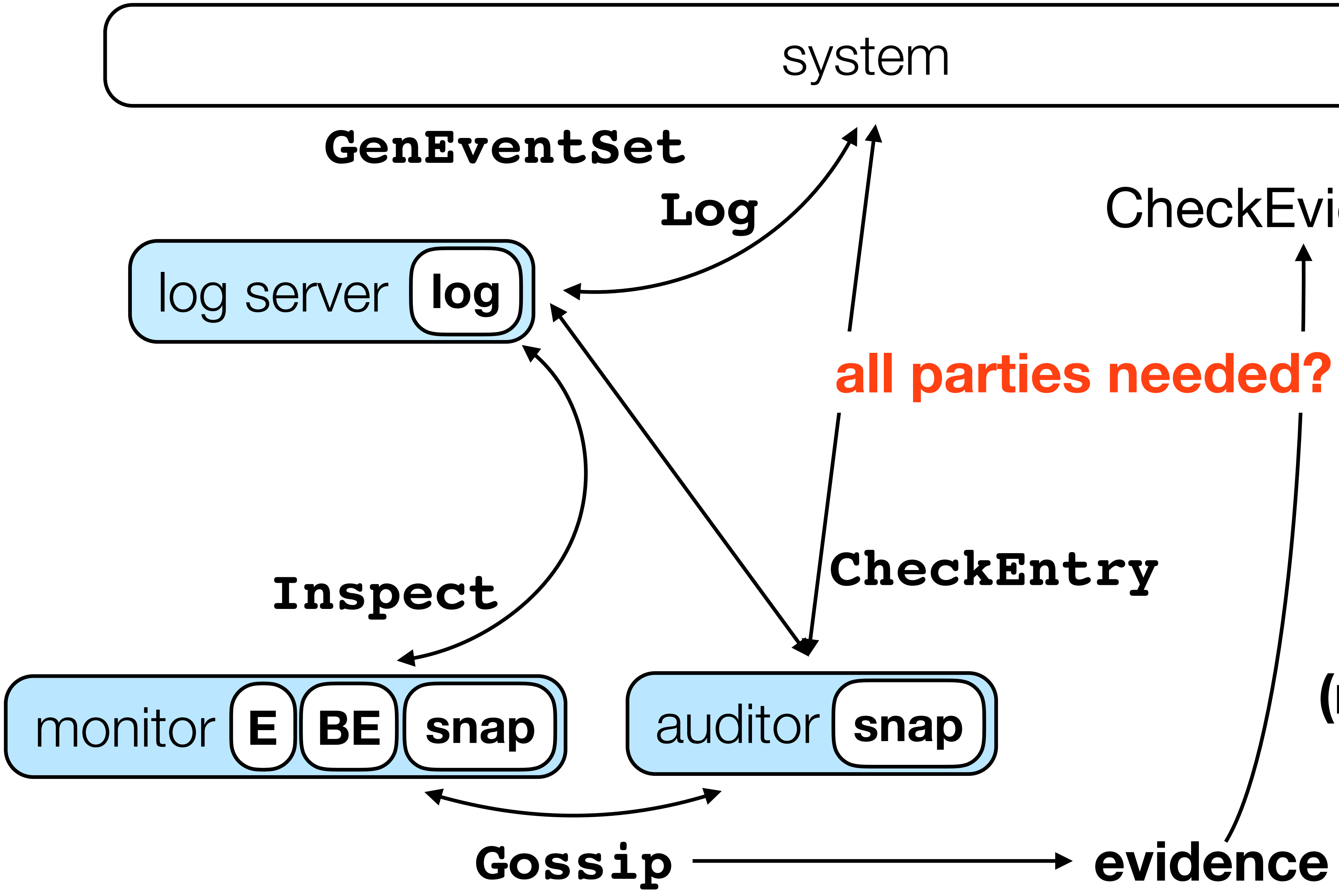
construction

(dlc+sig)

open problems

(CT+Bitcoin)

which systems?



system

GenEventSet

Log

CheckEvidence

log server **log**

all parties needed?

Inspect

CheckEntry

monitor **E** **BE** **snap**

auditor **snap**

Gossip

evidence

design

(add LS,Au,Mo)

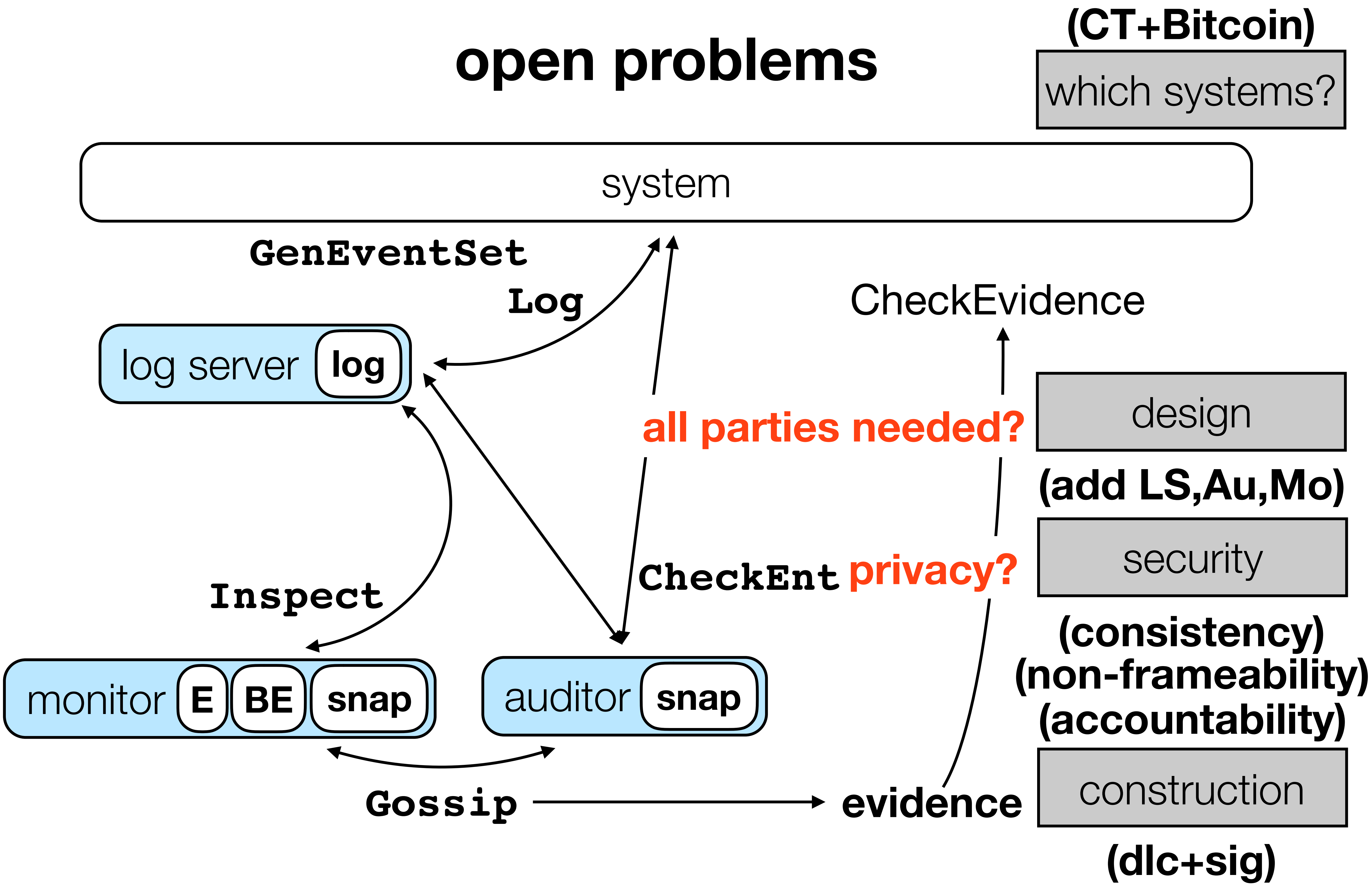
security

(consistency)
(non-frameability)
(accountability)

construction

(dlc+sig)

open problems



(CT+Bitcoin)

which systems?

system

GenEventSet

Log

CheckEvidence

log server

log

all parties needed?

design

(add LS,Au,Mo)

Inspect

CheckEnt

privacy?

security

(consistency)
(non-frameability)
(accountability)

monitor

E

BE

snap

auditor

snap

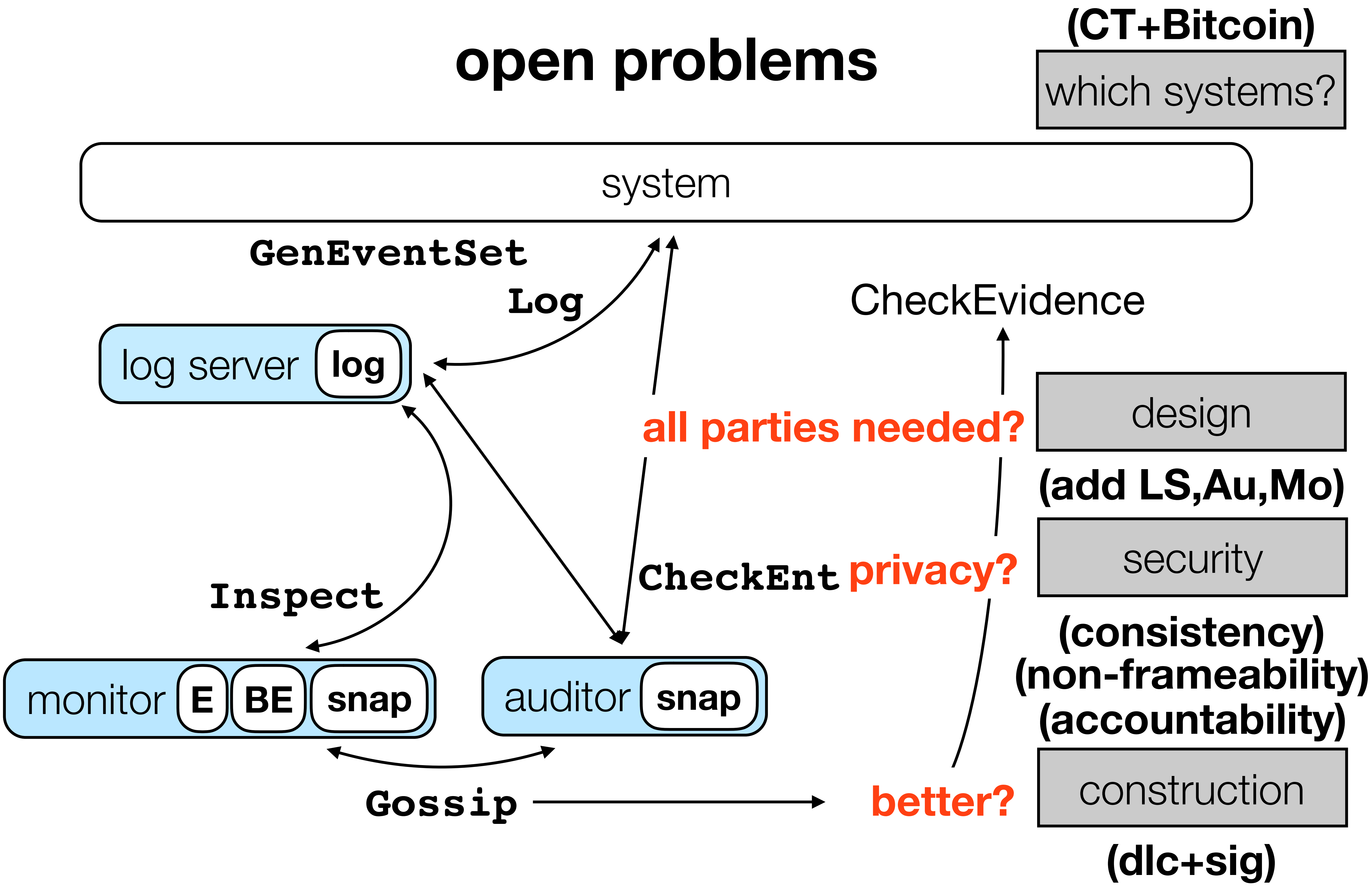
Gossip

evidence

construction

(dlc+sig)

open problems

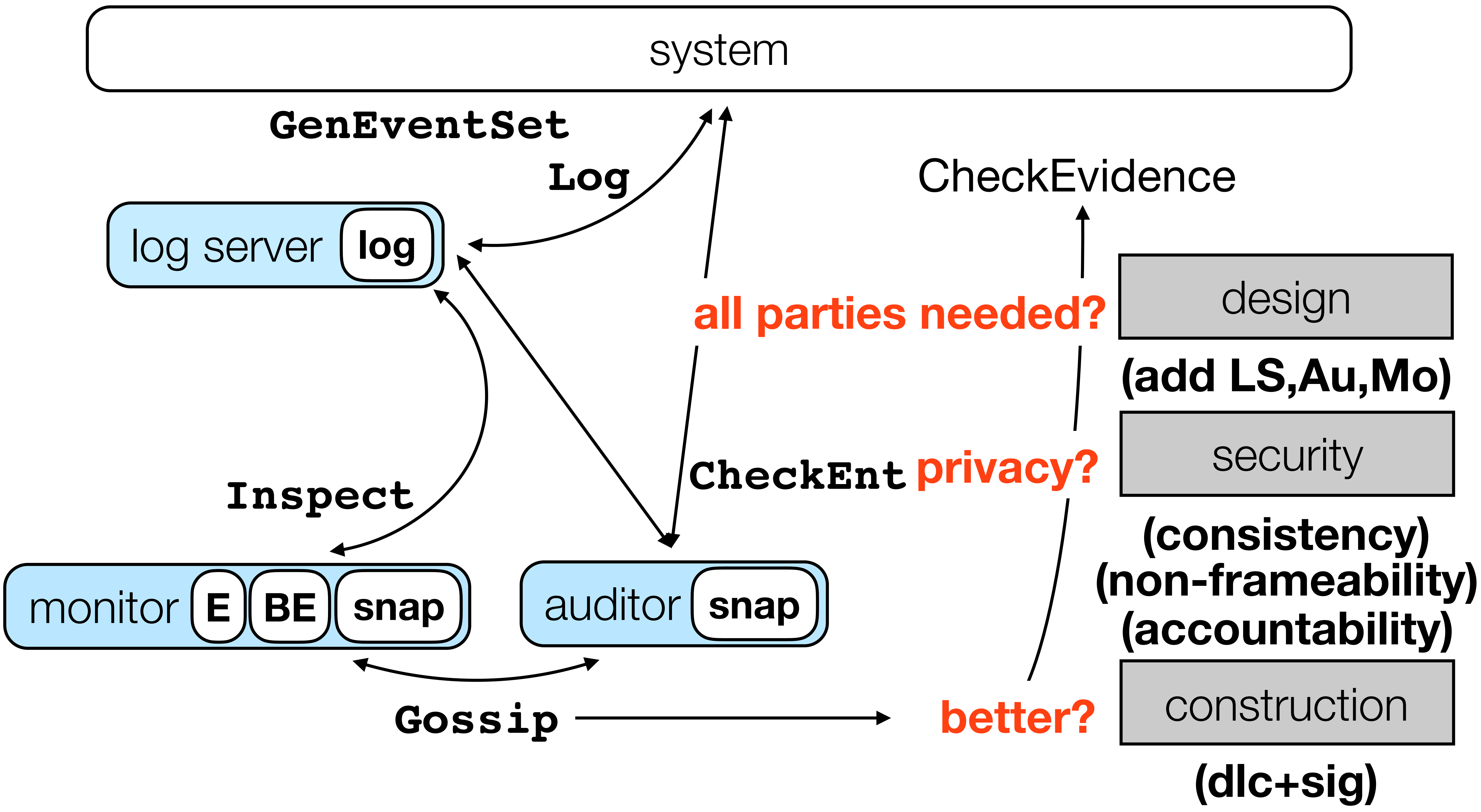


open problems

(CT+Bitcoin)

which systems?

others?



system

GenEventSet

Log

CheckEvidence

log server log

all parties needed?

design

(add LS,Au,Mo)

Inspect

CheckEnt

privacy?

security

(consistency)
(non-frameability)
(accountability)

monitor E BE snap

auditor snap

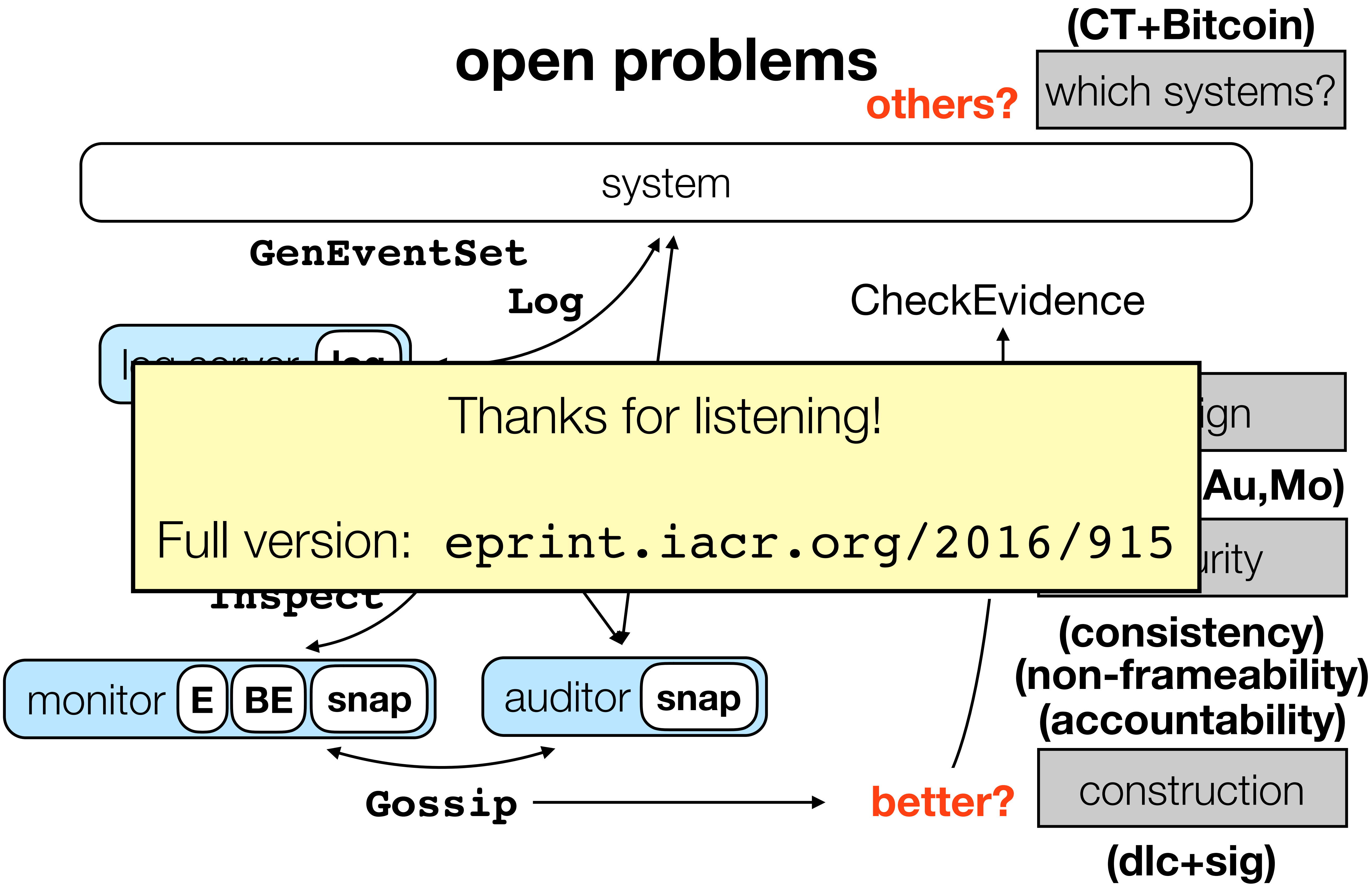
Gossip

better?

construction

(dlc+sig)

open problems



(CT+Bitcoin)

which systems?

others?

system

GenEventSet

Log

CheckEvidence

log server log

ign

Au,Mo)

urity

Inspect

monitor E BE snap

auditor snap

Gossip

better?

(consistency)
(non-frameability)
(accountability)

construction

(dlc+sig)

Thanks for listening!

Full version: `eprint.iacr.org/2016/915`

dynamic list commitment (dlc)

dynamic list commitment (dlc)

basic

Com

CheckCom

Append

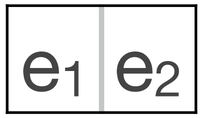
dynamic list commitment (dlc)

basic

Com

CheckCom

Append



dynamic list commitment (dlc)

basic

Com

CheckCom

Append

$$\boxed{e_1 \mid e_2}$$

$$\mathbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2||H(e_1))$$

dynamic list commitment (dlc)

basic

Com

CheckCom

Append

$\boxed{e_1 \mid e_2}$

$\textbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2 \| H(e_1))$

$\textbf{CheckCom}(c, \boxed{e_1 \mid e_2}) = (c = H(e_2 \| H(e_1)))$

dynamic list commitment (dlc)

basic

Com

CheckCom

Append

$$\boxed{e_1 \mid e_2}$$

$$\textbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2||H(e_1))$$

$$\textbf{CheckCom}(c, \boxed{e_1 \mid e_2}) = (c = H(e_2||H(e_1)))$$

$$\textbf{Append}(\boxed{e_3 \mid e_4}, c_{12}) = H(e_4||(H(e_3)||c_{12}))$$

dynamic list commitment (dlc)

basic

Com

CheckCom

Append

$$\boxed{e_1 \mid e_2}$$

$$\textbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2||H(e_1))$$

$$\textbf{CheckCom}(c, \boxed{e_1 \mid e_2}) = (c = H(e_2||H(e_1)))$$

$$\textbf{Append}(\boxed{e_3 \mid e_4}, c_{12}) = H(e_4||(H(e_3)||c_{12}))$$

$$\boxed{e_1 \mid e_2 \mid e_3 \mid e_4}$$

dynamic list commitment (dlc)

basic

Com
CheckCom
Append

all events?

ProveAppend
CheckAppend

$$\begin{array}{|c|c|}\hline e_1 & e_2 \\ \hline\end{array}$$

$$\textbf{Com}(\begin{array}{|c|c|}\hline e_1 & e_2 \\ \hline\end{array}) = H(e_2||H(e_1))$$

$$\textbf{CheckCom}(c,\begin{array}{|c|c|}\hline e_1 & e_2 \\ \hline\end{array}) = (c = H(e_2||H(e_1)))$$

$$\textbf{Append}(\begin{array}{|c|c|}\hline e_3 & e_4 \\ \hline\end{array},c_{12}) = H(e_4||(H(e_3)||c_{12}))$$

$$\begin{array}{|c|c|c|c|}\hline e_1 & e_2 & e_3 & e_4 \\ \hline\end{array}$$

dynamic list commitment (dlc)

basic

Com
CheckCom
Append

all events?

ProveAppend
CheckAppend

$$\boxed{e_1 \mid e_2}$$

Com$(\boxed{e_1 \mid e_2}) = H(e_2||H(e_1))$

CheckCom$(c, \boxed{e_1 \mid e_2}) = (c = H(e_2||H(e_1)))$

Append$(\boxed{e_3 \mid e_4}, c_{12}) = H(e_4||(H(e_3)||c_{12}))$

$$\boxed{e_1 \mid e_2 \mid e_3 \mid e_4}$$

ProveAppend$(c_{12}, c_{1234}, \boxed{e_1 \mid e_2 \mid e_3 \mid e_4}) = \boxed{e_3 \mid e_4}$

dynamic list commitment (dlc)

basic

Com
CheckCom
Append

all events?

ProveAppend
CheckAppend

$$\boxed{e_1 \mid e_2}$$

$$\textbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2 \| H(e_1))$$

$$\textbf{CheckCom}(c, \boxed{e_1 \mid e_2}) = (c = H(e_2 \| H(e_1)))$$

$$\textbf{Append}(\boxed{e_3 \mid e_4}, c_{12}) = H(e_4 \| (H(e_3) \| c_{12}))$$

$$\boxed{e_1 \mid e_2 \mid e_3 \mid e_4}$$

$$\textbf{ProveAppend}(c_{12}, c_{1234}, \boxed{e_1 \mid e_2 \mid e_3 \mid e_4}) = \boxed{e_3 \mid e_4}$$

$$\textbf{CheckAppend}(c_{12}, c_{1234}, \boxed{e_3 \mid e_4}) = (c_{1234} = \text{Append}(\boxed{e_3 \mid e_4}, c_{12}))$$

dynamic list commitment (dlc)

basic

Com
CheckCom
Append

all events?

ProveAppend
CheckAppend

specific event?

ProveIncl
CheckIncl

$$\boxed{e_1 \mid e_2}$$

$$\textbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2||H(e_1))$$

$$\textbf{CheckCom}(c,\boxed{e_1 \mid e_2}) = (c = H(e_2||H(e_1)))$$

$$\textbf{Append}(\boxed{e_3 \mid e_4},c_{12}) = H(e_4||(H(e_3)||c_{12}))$$

$$\boxed{e_1 \mid e_2 \mid e_3 \mid e_4}$$

$$\textbf{ProveAppend}(c_{12},c_{1234},\boxed{e_1 \mid e_2 \mid e_3 \mid e_4}) = \boxed{e_3 \mid e_4}$$

$$\textbf{CheckAppend}(c_{12},c_{1234},\boxed{e_3 \mid e_4}) = (c_{1234} = \text{Append}(\boxed{e_3 \mid e_4},c_{12}))$$

dynamic list commitment (dlc)

basic	all events?	specific event?
Com	**ProveAppend**	**ProveIncl**
CheckCom	**CheckAppend**	**CheckIncl**
Append		

$$\boxed{e_1 \mid e_2}$$

$$\textbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2 || H(e_1))$$

$$\textbf{CheckCom}(c, \boxed{e_1 \mid e_2}) = (c = H(e_2 || H(e_1)))$$

$$\textbf{Append}(\boxed{e_3 \mid e_4}, c_{12}) = H(e_4 || (H(e_3) || c_{12}))$$

$$\boxed{e_1 \mid e_2 \mid e_3 \mid e_4}$$

$$\textbf{ProveAppend}(c_{12}, c_{1234}, \boxed{e_1 \mid e_2 \mid e_3 \mid e_4}) = \boxed{e_3 \mid e_4}$$

$$\textbf{CheckAppend}(c_{12}, c_{1234}, \boxed{e_3 \mid e_4}) = (c_{1234} = \text{Append}(\boxed{e_3 \mid e_4}, c_{12}))$$

$$\textbf{ProveIncl}(c_{1234}, e_3, \boxed{e_1 \mid e_2 \mid e_3 \mid e_4}) = (c_{12}, \boxed{e_4})$$

dynamic list commitment (dlc)

basic	all events?	specific event?
Com	**ProveAppend**	**ProveIncl**
CheckCom	**CheckAppend**	**CheckIncl**
Append		

$$\boxed{e_1 \mid e_2}$$

$$\textbf{Com}(\boxed{e_1 \mid e_2}) = H(e_2||H(e_1))$$

$$\textbf{CheckCom}(c, \boxed{e_1 \mid e_2}) = (c = H(e_2||H(e_1)))$$

$$\textbf{Append}(\boxed{e_3 \mid e_4}, c_{12}) = H(e_4||(H(e_3)||c_{12}))$$

$$\boxed{e_1 \mid e_2 \mid e_3 \mid e_4}$$

$$\textbf{ProveAppend}(c_{12}, c_{1234}, \boxed{e_1 \mid e_2 \mid e_3 \mid e_4}) = \boxed{e_3 \mid e_4}$$

$$\textbf{CheckAppend}(c_{12}, c_{1234}, \boxed{e_3 \mid e_4}) = (c_{1234} = \text{Append}(\boxed{e_3 \mid e_4}, c_{12}))$$

$$\textbf{ProveIncl}(c_{1234}, e_3, \boxed{e_1 \mid e_2 \mid e_3 \mid e_4}) = (c_{12}, \boxed{e_4})$$

$$\textbf{CheckIncl}(c_{1234}, e_3, (c_{12}, \boxed{e_4})) = \text{CheckAppend}(c_{12}, c_{1234}, \boxed{e_3 \mid e_4})$$