Gateway Crash Recovery Overview

IETF 112 - November, 12th, 2021

draft-belchior-blockchain-gateway-recovery-03

R. Belchior (Técnico Lisboa)

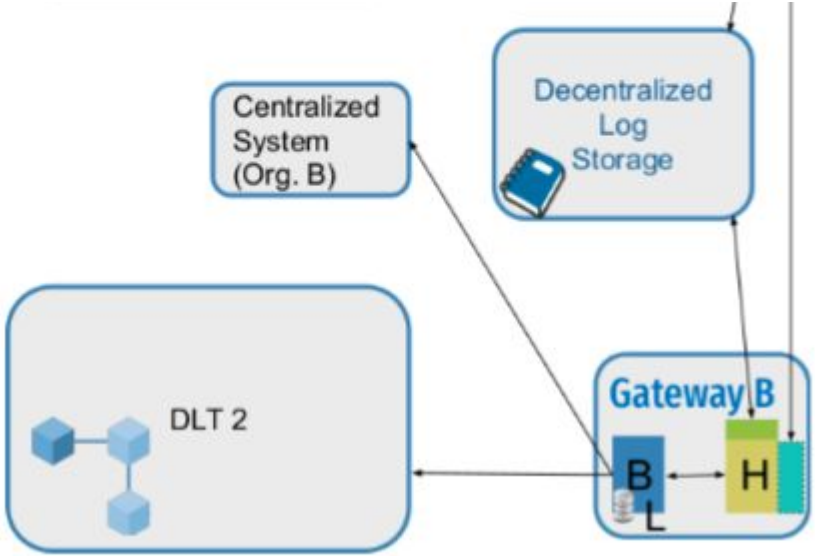
rafael.belchior@tecnico.ulisboa.pt

M. Correia (Técnico Lisboa)

T. Hardjono (MIT)

TÉCNICO LISBOA

inesc id lisboa 20 YEARS DEFINING TECHNOLOGY



Source:
https://www.techrxiv.org/articles/preprint/HERMES_Fault-Tolerant_Middleware_for_Blockchain_Interoperability/14120291

- Crash recovery of gateway sessions provides desirable properties to transactions originated by a gateway-to-gateway session.

Draft v03

https://datatracker.ietf.org/doc/draft-belchior-gateway-recovery/03/

- Relies on logging and the assumption of recovering (self-healing or primary-backup)

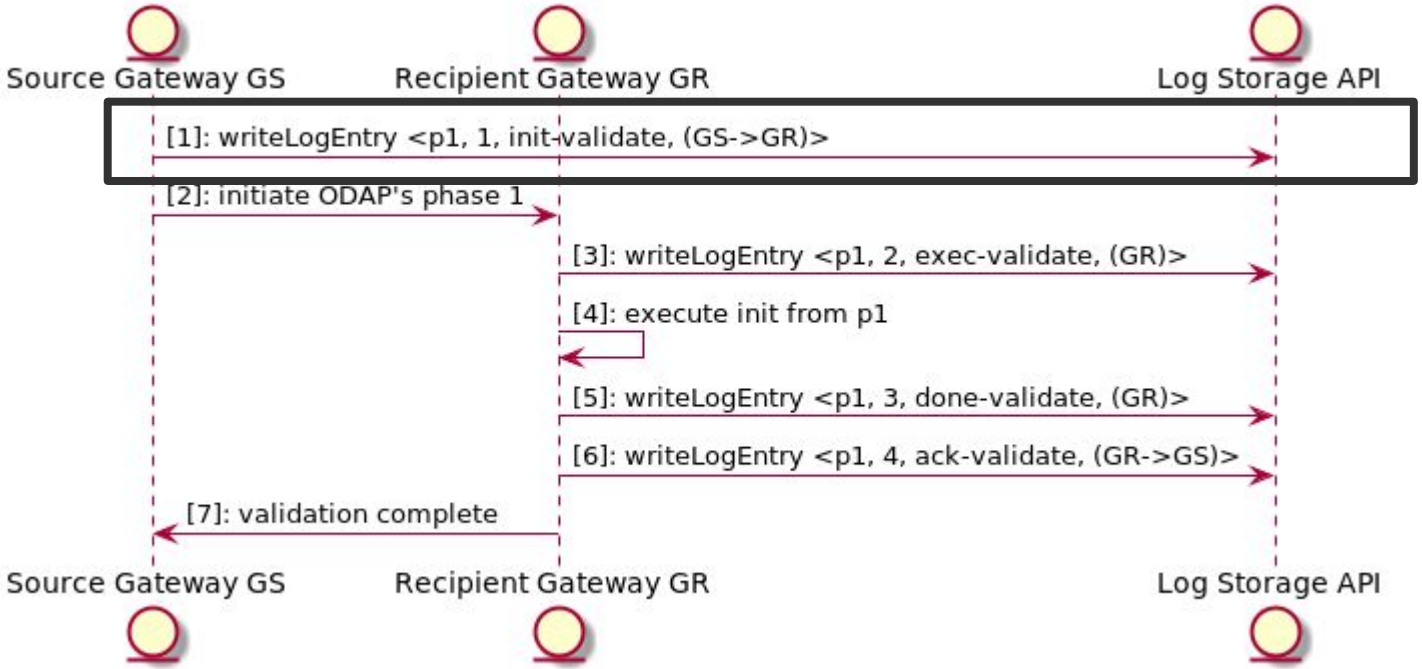
- Recovery is done on every phase of ODAP, with a “specialized procedure” on the Commitment Establishment Flow

Message and logging flow

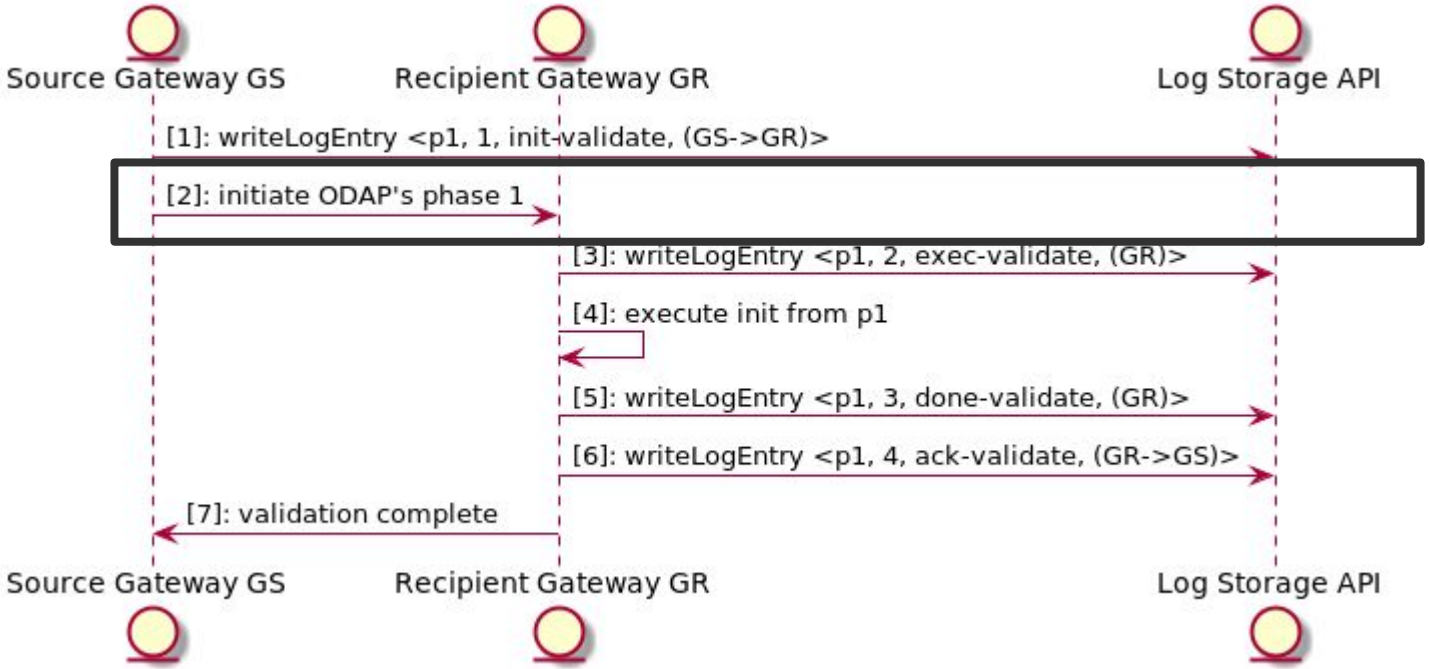
For each gateway

Message and logging flow

init-

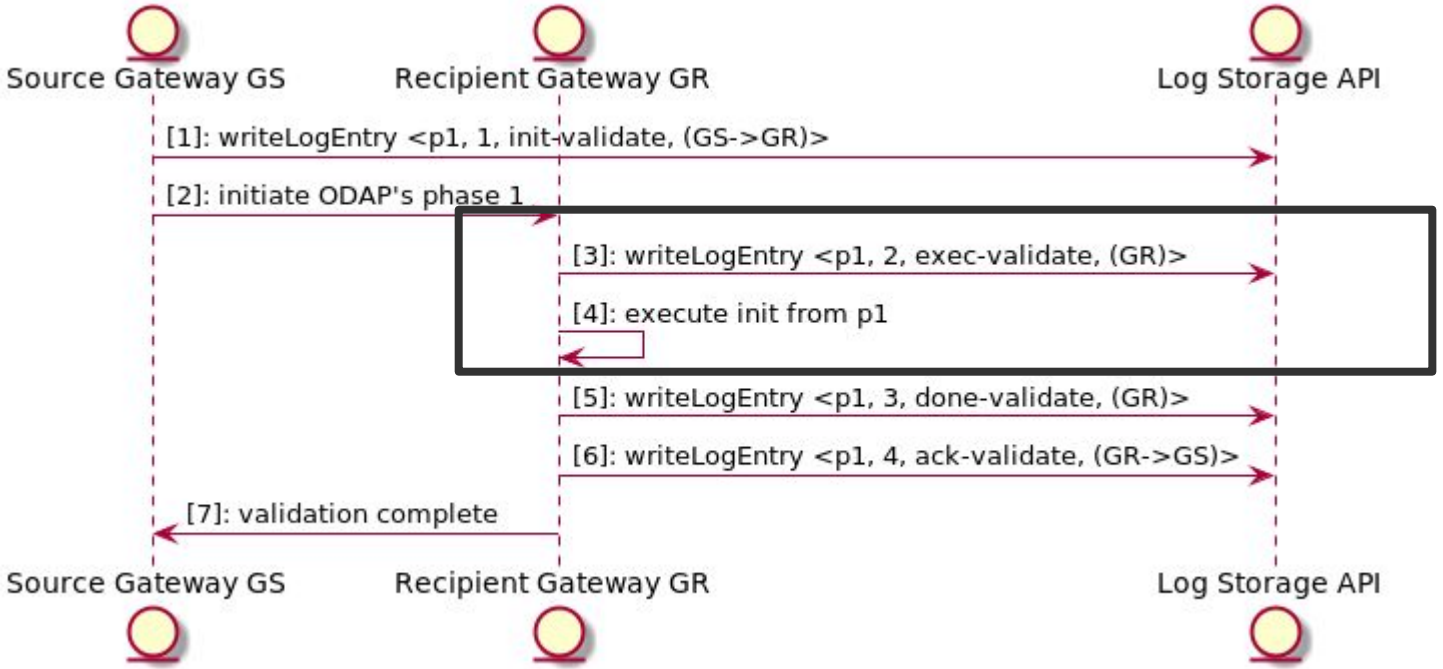


Message and logging flow



Message and logging flow

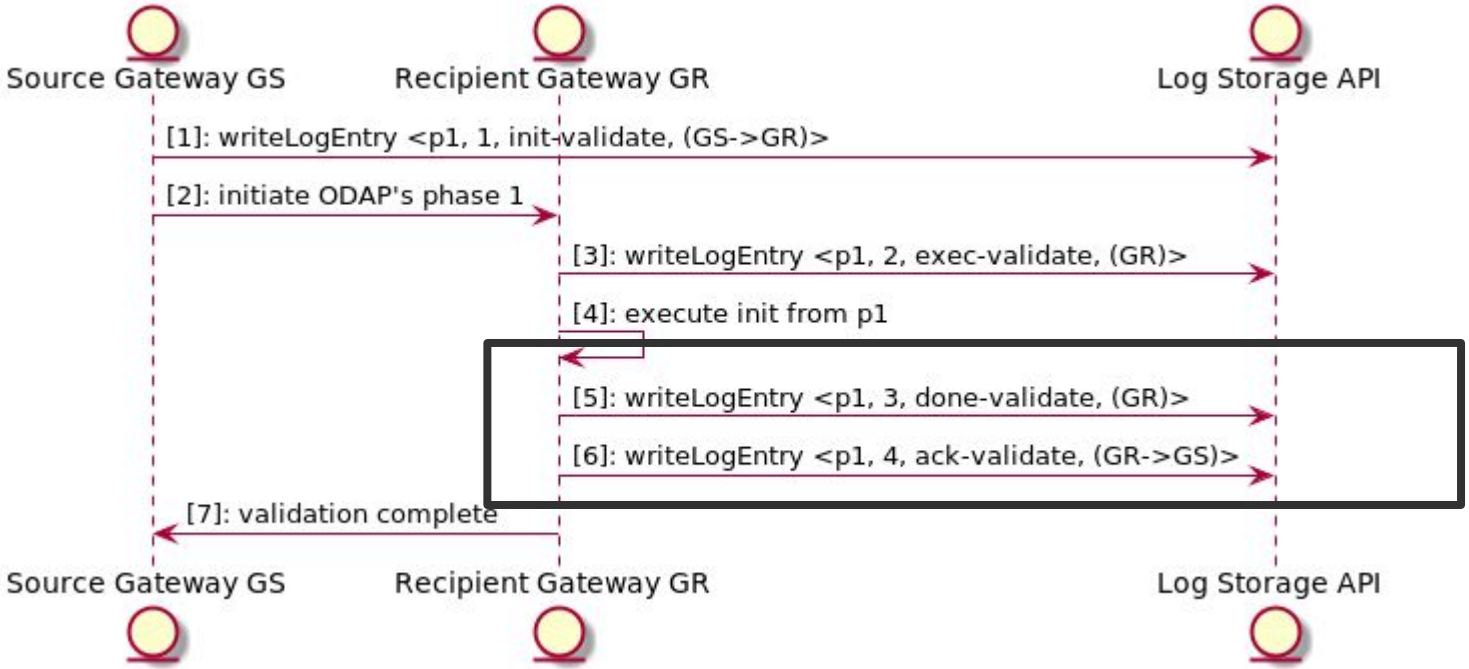
exec-



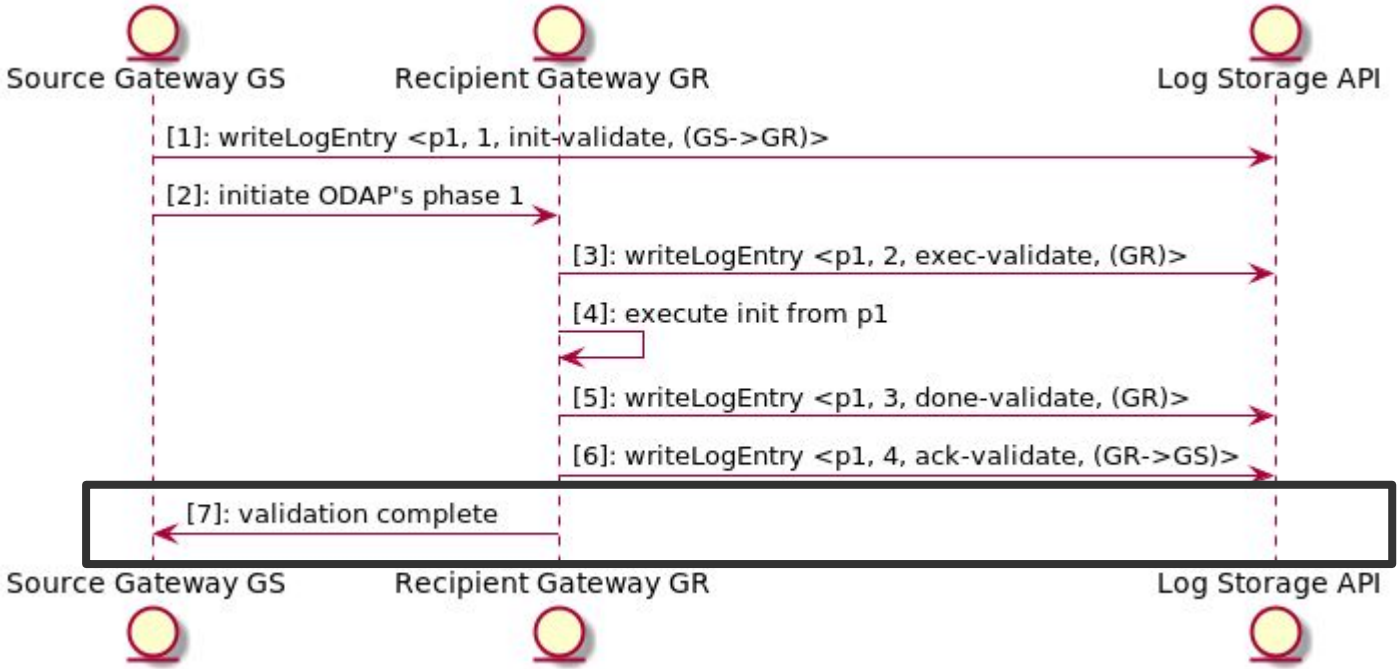
Message and logging flow

done-

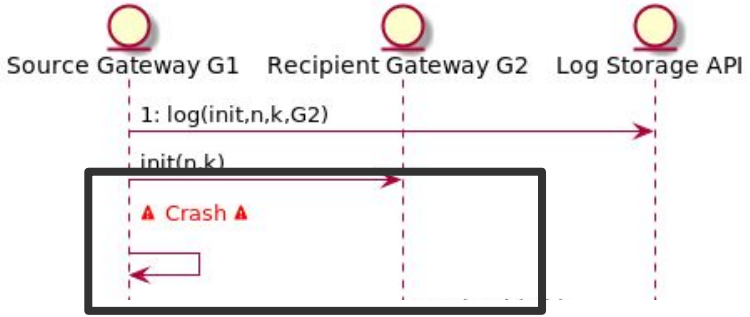
ack-



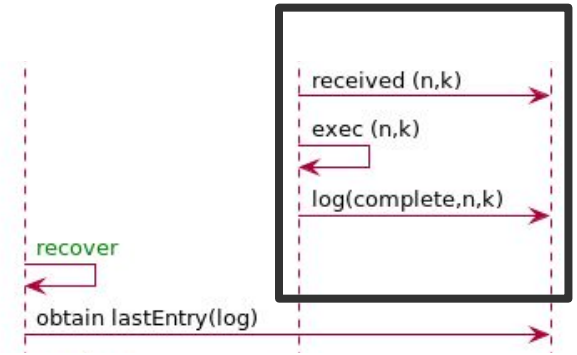
Message and logging flow



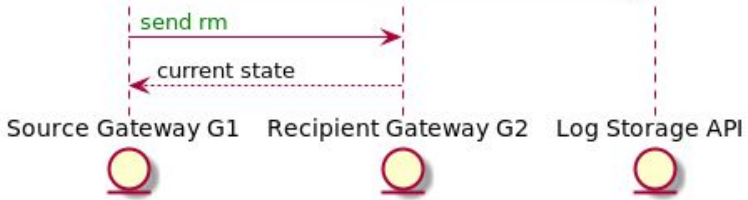
Recovery Procedure



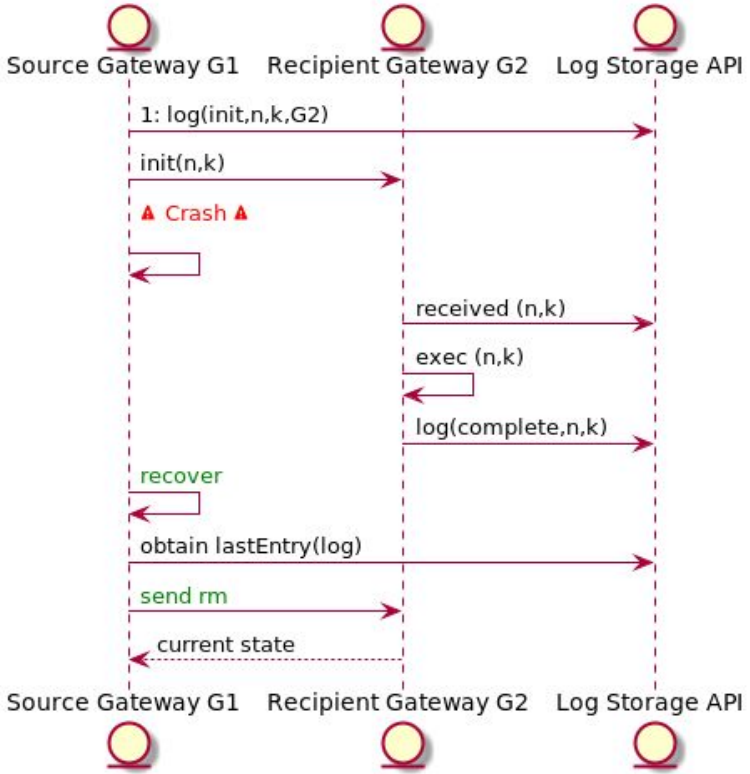
Recovery Procedure



Recovery Procedure

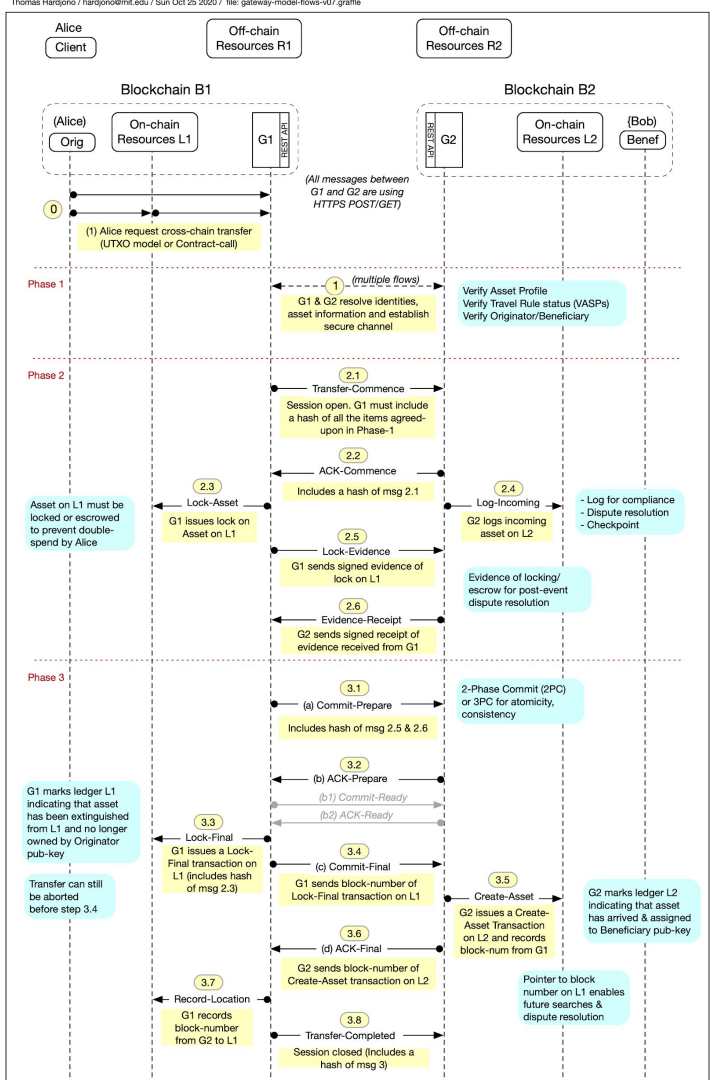


Recovery Procedure



Message & Logging Flow

ODAP-2PC



Algorithm 1: ODAP-2PC Protocol

Input: Coordinator $\mathcal{G}_S$, Participant $\mathcal{G}_R$, Asset a, Gateway primitives PRE_LOCK,LOCK,COMMIT,CREATE_ASSET, COMPLETE, ROLLBACK

Result: Asset a transferred from $\mathcal{G}_S$ to $\mathcal{G}_R$

1 $PO_{\mathcal{G}_S} = \perp$ ▷ operations to be rolledback in case of failure for $\mathcal{G}_S$

2 $PO_{\mathcal{G}_R} = \perp$ ▷ operations to be rolledback in case of failure for $\mathcal{G}_R$

3 preLock $= \mathcal{G}_S$.PRE_LOCK(a) ▷ step 2.3

Draft v03

https://datatracker.ietf.org/doc/draft-belchior-gateway-recovery/03/

Academic Paper

https://bit.ly/3GzVOjY

DLT Gateway Crash Recovery

IETF 112 - November, 12th, 2021

draft-belchior-blockchain-gateway-recovery-03

R. Belchior (Técnico Lisboa)

rafael.belchior@tecnico.ulisboa.pt

M. Correia (Técnico Lisboa)

T. Hardjono (MIT)

TÉCNICO LISBOA

inesc id lisboa 20 YEARS DEFINING TECHNOLOGY

Algorithm 1: ODAP-2PC Protocol

Input: Coordinator $\mathcal{G}_S$, Participant $\mathcal{G}_R$, Asset a, Gateway primitives PRE_LOCK,LOCK,COMMIT,CREATE_ASSET, COMPLETE, ROLLBACK

Result: Asset a transferred from $\mathcal{G}_S$ to $\mathcal{G}_R$

1 $PO_{\mathcal{G}_S} = \perp$ ▷ operations to be rolledback in case of failure for $\mathcal{G}_S$

2 $PO_{\mathcal{G}_R} = \perp$ ▷ operations to be rolledback in case of failure for $\mathcal{G}_R$

3 preLock $= \mathcal{G}_S$.PRE_LOCK(a) ▷ step 2.3

5 ▷ Voting Phase

6 $\mathcal{G}_S \overset{vote-req}{\longrightarrow} \mathcal{G}_R$ ▷ step 3.1

7 **wait until** $\mathcal{G}_R \overset{\alpha(vote-req)}{\longrightarrow} \mathcal{G}_S$ ▷ step 3.2

8 ▷ Decision Phase

9 **if** $\mathcal{G}_R \overset{\alpha(vote-req)}{\longrightarrow} \mathcal{G}_S = NO$ **then**

10 $\quad \mathcal{G}_S \overset{abort()}{\longrightarrow} \mathcal{G}_R$ ▷ otherwise, $\mathcal{G}_R \overset{\alpha(vote-req)}{\longrightarrow} \mathcal{G}_S = \text{YES}$

11 $\quad \mathcal{G}_S$.ROLLBACK($PO_{\mathcal{G}_S}$) ▷ undo $\mathcal{G}_S.preLock(a)$

13 $lock = \mathcal{G}_S$.LOCK(a) ▷ step 3.3

14 $PO_{\mathcal{G}_S}$.append(lock)

15 $commit = \mathcal{G}_S$.COMMIT() ▷ step 3.4

16 **if** $commit = \perp$ **then**

17 $\quad \mathcal{G}_S \xrightarrow{abort()} \mathcal{G}_R$

18 $\quad \mathcal{G}_S.rollback(PO_{\mathcal{G}_S})$ ▷ undo $\mathcal{G}_S$.LOCK(a)

19 **end if**

20 $\mathcal{G}_S \xrightarrow{commit} \mathcal{G}_R$

21 $a' = \mathcal{G}_R$.CREATE_ASSET() ▷ step 3.5, corresponds to the asset transfer

22 $PO_{\mathcal{G}_R}$.append(a')

```
23 wait until G_R --α(commit)--> G_S                          ▷ step 3.6
24 if G_R --α(commit)--> G_S = COMMIT then
25 |  G_S.COMPLETE()                                          ▷ step 3.8
26 end if
27 else
28 |  G_S --abort()--> G_R                  ▷ otherwise, G_R failed the commit
29 |  G_S.ROLLBACK(PO_{G_S})                                  ▷ undo G_S locks
30 |  G_R.ROLLBACK(PO_{G_R})                ▷ undo G_R.CREATE_ASSET()
31 end if
32 return                                                     ▷ asset transferred
```