

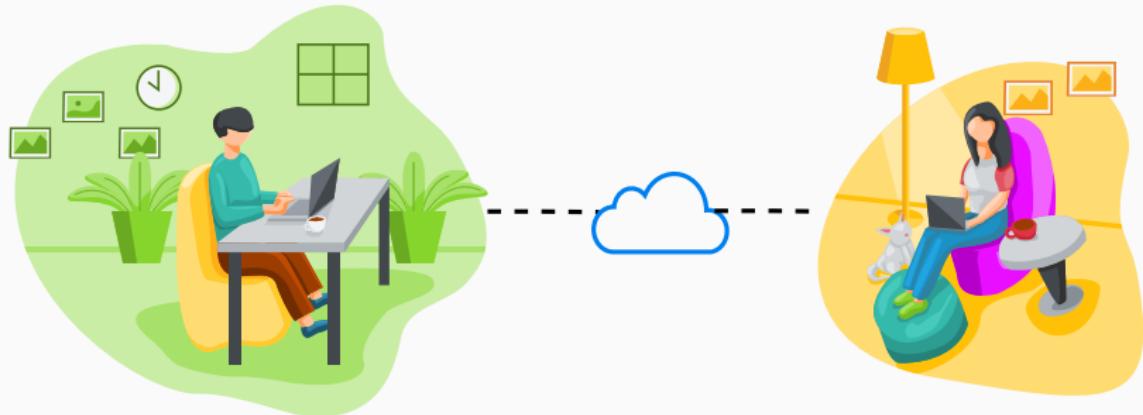
Synql: A CRDT-based Approach for Replicated Relational Databases with Integrity Constraints

Victorien Elvinger, Claudia-Lavinia Ignat, Habibatou Ba
Inria Nancy, France



June 2024

Adding collaborative features to applications



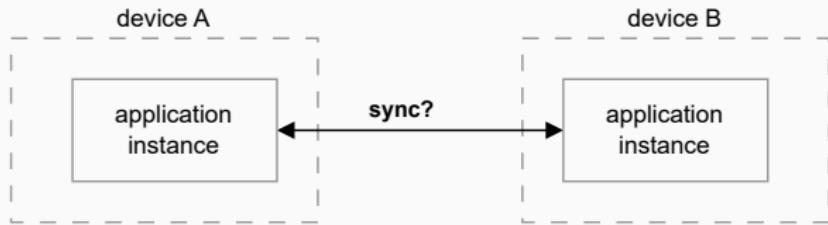
- several users **modify together** a shared content
 - located at **different places**
 - **simultaneous** modifications or at **distinct time**
- adding collaborative features to applications is hard
 - **sequential** to **concurrent** modifications

Adding collaborative features to applications



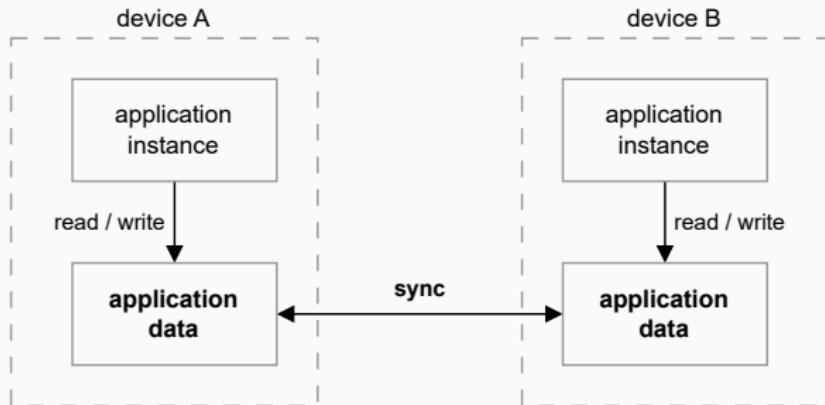
- several users **modify together** a shared content
 - located at **different places**
 - **simultaneous** modifications or at **distinct time**
- adding collaborative features to applications is hard
 - **sequential to concurrent** modifications
 - **offline support**

Adding collaborative features to applications



- replicate the application \implies dedicated development

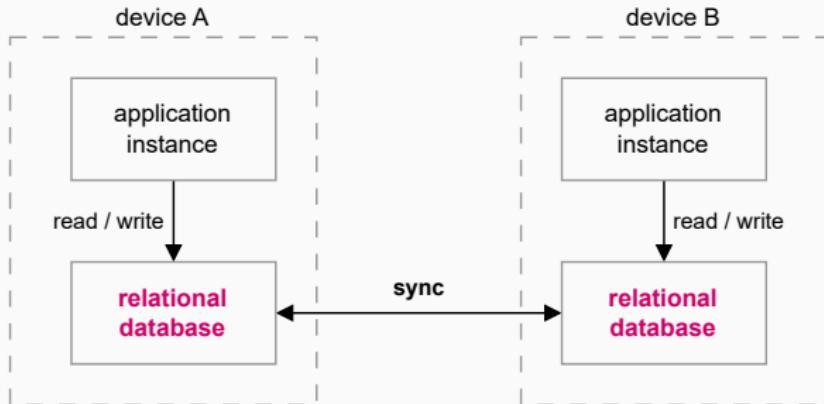
Adding collaborative features to applications



- replicate the application \implies dedicated development
- replicate the application data^a

^aKleppmann et al., “Local-first software: you own your data, in spite of the cloud”, 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Onward!, Athens, Greece, 2019.

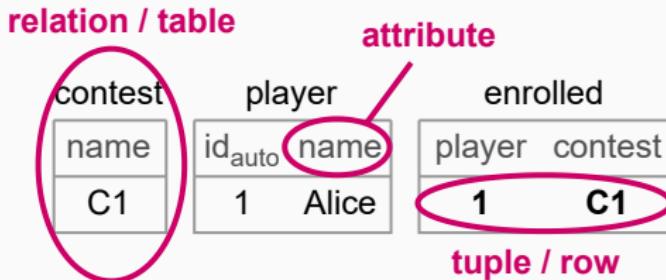
Adding collaborative features to applications



- replicate the application \implies dedicated development
- replicate the application data^a
- many applications embed a relational database

^aKleppmann et al., “Local-first software: you own your data, in spite of the cloud”, 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Onward!, Athens, Greece, 2019.

Relational databases and integrity constraints



Relational databases and integrity constraints

contest	player	enrolled
name	id _{auto} name	player contest
C1	1 Alice	1 C1
	2 Bea	2 C1

uniqueness integrity

- uniqueness integrity: relation's tuples are uniquely identified

Relational databases and integrity constraints



- uniqueness integrity: relation's tuples are **uniquely identified**
- referential integrity: **target of a reference exists**

Relational databases and integrity constraints



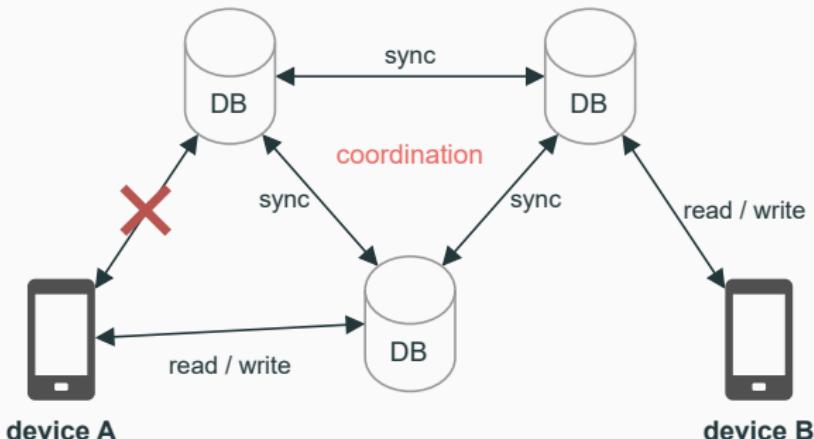
- uniqueness integrity: relation's tuples are **uniquely identified**
- referential integrity: **target of a reference exists**
- the deletion of a reference target can result in
 - the **cascading of the deletion** to the reference source

Relational databases and integrity constraints



- uniqueness integrity: relation's tuples are **uniquely identified**
- referential integrity: **target of a reference exists**
- the deletion of a reference target can result in
 - the **cascading of the deletion** to the reference source
 - the **abortion of the deletion**

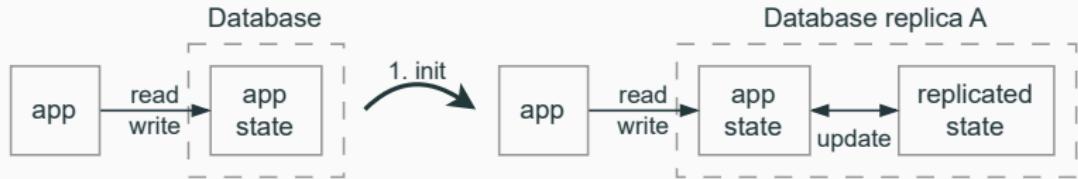
Replicating relational databases: already done?



- client-server architecture
- coordination to maintain data integrity^a

^aBailis et al., “Highly Available Transactions: Virtues and Limitations”, 2013.

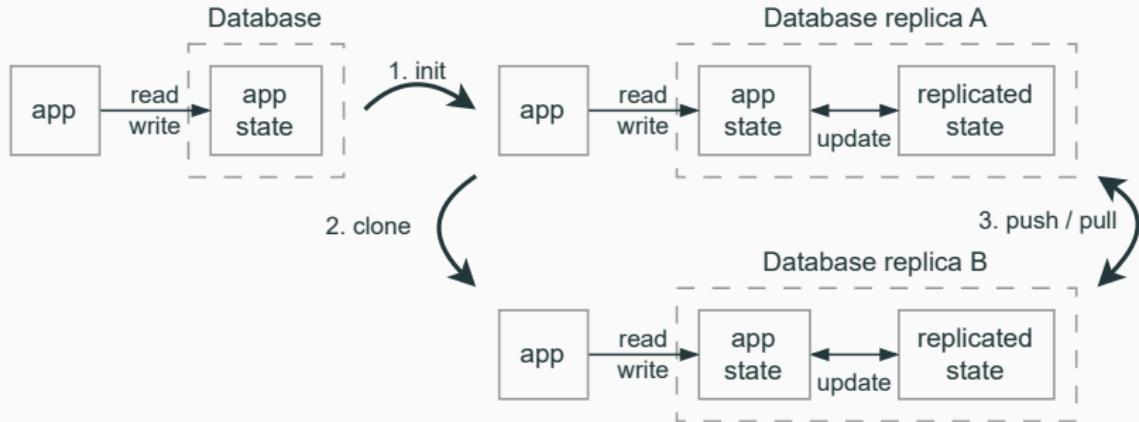
Coordination-less replication of relational databases



- Git-like **coordination-less** replication of relational databases^a
- can **break data integrity and user intent**
- **not Strongly Convergent**

^aYu et al., “Conflict-Free Replicated Relations for Multi-Synchronous Database Management at Edge”, *IEEE International Conference on Smart Data Services SMDS, Beijing, China, 2020.*

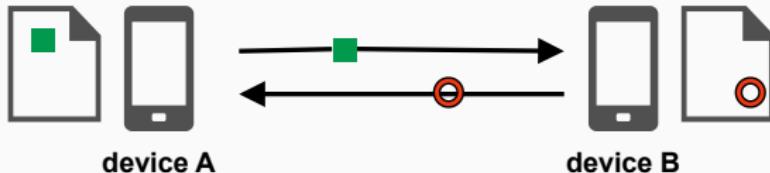
Coordination-less replication of relational databases



- Git-like **coordination-less** replication of relational databases^a
- can **break data integrity and user intent**
- **not Strongly Convergent**

^aYu et al., “Conflict-Free Replicated Relations for Multi-Synchronous Database Management at Edge”, *IEEE International Conference on Smart Data Services SMDS, Beijing, China, 2020.*

Strong convergence



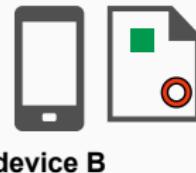
- if two devices integrate the same modifications,
then they converge to an identical state^a
- advantages:
 - low latency
 - no flickering

^aShapiro et al., "Conflict-Free Replicated Data Types", *Stabilization, Safety, and Security of Distributed Systems - 13th International Symposium SSS, Grenoble, France, 2011*, 6/15

Strong convergence



device A

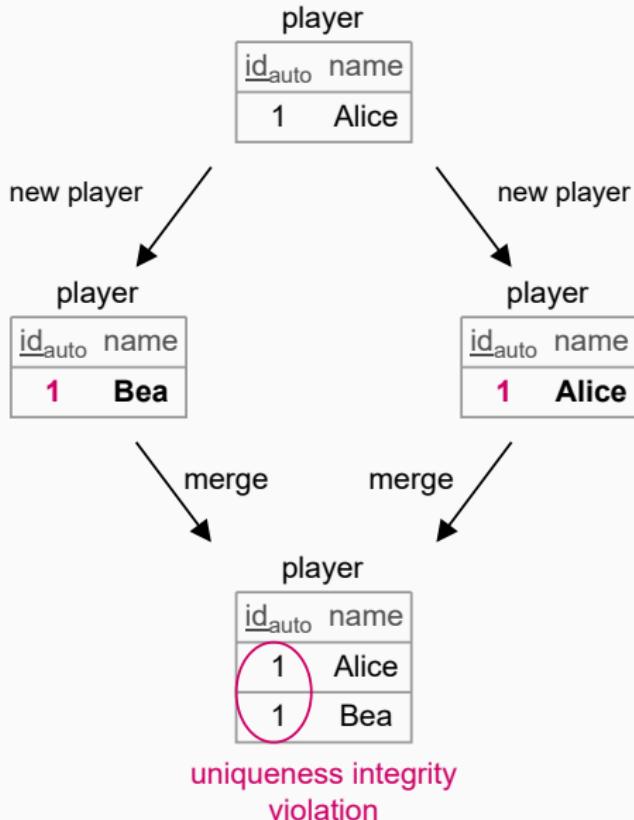


device B

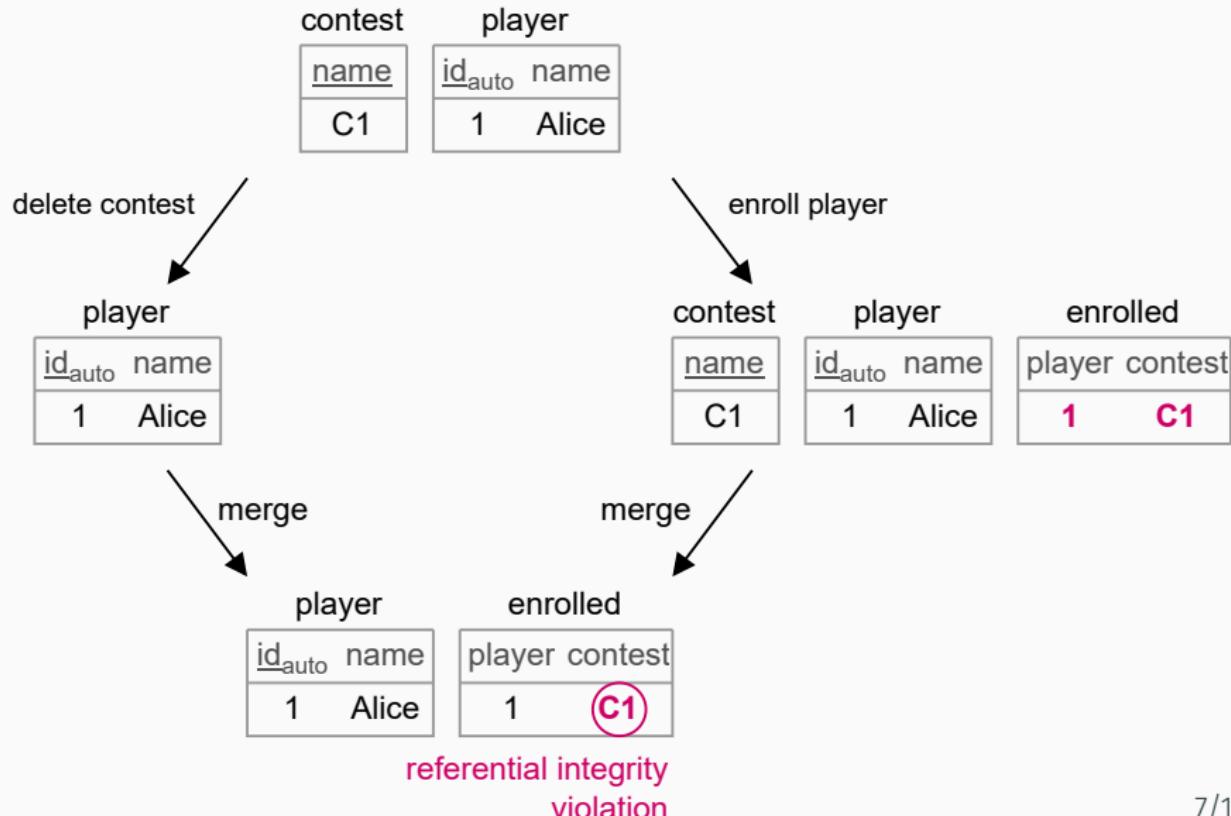
- if two devices integrate the same modifications, then they converge to an identical state^a
- advantages:
 - low latency
 - no flickering

^aShapiro et al., "Conflict-Free Replicated Data Types", *Stabilization, Safety, and Security of Distributed Systems - 13th International Symposium SSS, Grenoble, France, 2011*. 6/15

Data integrity in face of coordination-less concurrencies

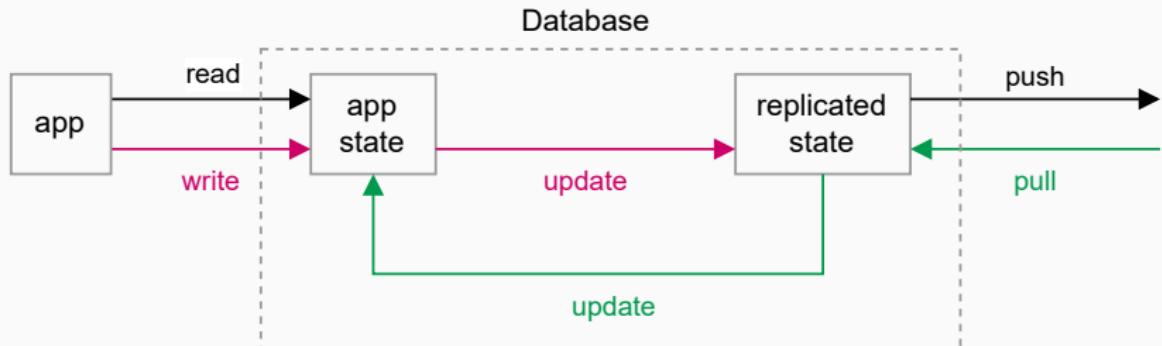


Data integrity in face of coordination-less concurrencies



Can we replicate a relational database without
any coordination that enforces Strong
Convergence and maintains data integrity?

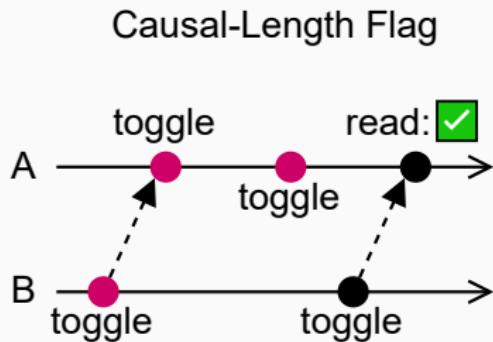
Architecture overview



- app read without overhead
- an app write triggers replicated state update
- push / pull in background
- a pull merges the received state and updates the app state

Conflict-free Replicated Data Types¹ (CRDTs)

- CRDTs allow concurrent edits without coordination
- specified as Abstract Data Types with a **merge** function
- enforce **Strong Convergence**



$$\text{CLFlag} \stackrel{\text{def}}{=} \mathbb{N}_0 \quad (1)$$

$$\text{read}(n) \stackrel{\text{def}}{=} \text{isOdd}(n) \quad (2)$$

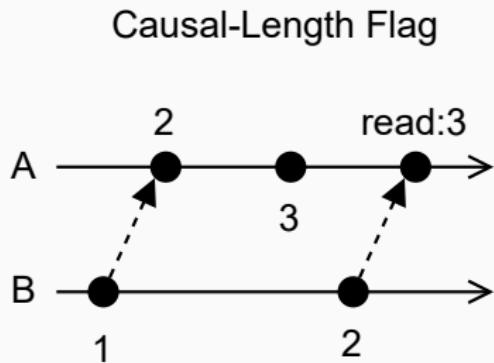
$$\text{toggle}(n) \stackrel{\text{def}}{=} n + 1 \quad (3)$$

$$n \sqcup n' \stackrel{\text{def}}{=} \max(n, n') \quad (4)$$

¹Shapiro et al., “Conflict-Free Replicated Data Types”, *Stabilization, Safety, and Security of Distributed Systems - 13th International Symposium SSS, Grenoble, France, 2011.*

Conflict-free Replicated Data Types¹ (CRDTs)

- CRDTs allow concurrent edits without coordination
- specified as Abstract Data Types with a **merge** function
- enforce **Strong Convergence**



$$\text{CLFlag} \stackrel{\text{def}}{=} \mathbb{N}_0 \quad (1)$$

$$\text{read}(n) \stackrel{\text{def}}{=} \text{isOdd}(n) \quad (2)$$

$$\text{toggle}(n) \stackrel{\text{def}}{=} n + 1 \quad (3)$$

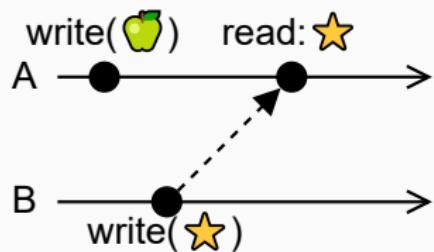
$$n \sqcup n' \stackrel{\text{def}}{=} \max(n, n') \quad (4)$$

¹Shapiro et al., "Conflict-Free Replicated Data Types", *Stabilization, Safety, and Security of Distributed Systems - 13th International Symposium SSS, Grenoble, France, 2011.*

Conflict-free Replicated Data Types¹ (CRDTs)

- CRDTs allow concurrent edits without coordination
- specified as Abstract Data Types with a **merge** function
- enforce **Strong Convergence**

Last-Writer-Win Register



$$\text{LWWReg} \stackrel{\text{def}}{=} \text{Value} \times \text{Ts} \quad (1)$$

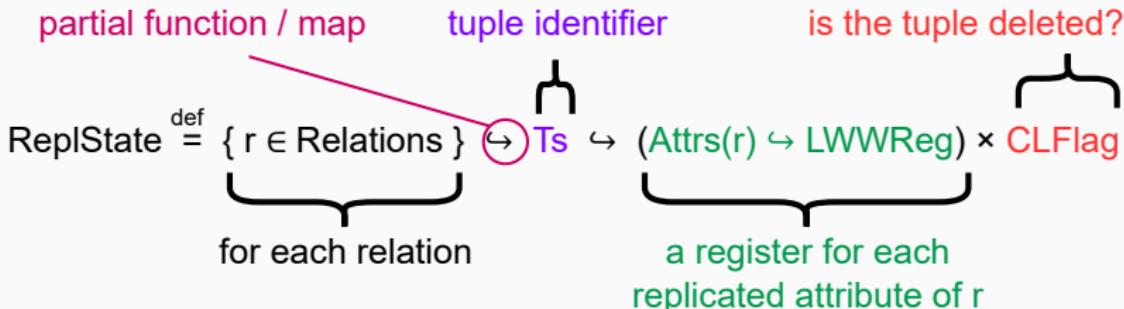
$$\text{read}(\langle v, t \rangle) \stackrel{\text{def}}{=} v \quad (2)$$

$$\text{write}(v) \stackrel{\text{def}}{=} \langle v, \text{new_ts} \rangle \quad (3)$$

$$\langle v, t \rangle \sqcup \langle v', t' \rangle \stackrel{\text{def}}{=} \begin{cases} \langle v, t \rangle & \text{if } t > t' \\ \langle v', t' \rangle & \end{cases} \quad (4)$$

¹Shapiro et al., “Conflict-Free Replicated Data Types”, *Stabilization, Safety, and Security of Distributed Systems - 13th International Symposium SSS, Grenoble, France, 2011.*

Replicated state: composing CRDTs



- the composition^a of CRDTs form a new CRDT
- timestamps (Ts) are **monotonic** and **globally unique**

^aBaquero et al., “Composition in State-based Replicated Data Types”, 2017.

Replicated state: composing CRDTs

partial function / map tuple identifier is the tuple deleted?

$\text{ReplState} \stackrel{\text{def}}{=} \{ r \in \text{Relations} \}$

for each relation $Ts \hookrightarrow (\text{Attrs}(r) \hookleftarrow \text{LWWReg}) \times \text{CLFlag}$

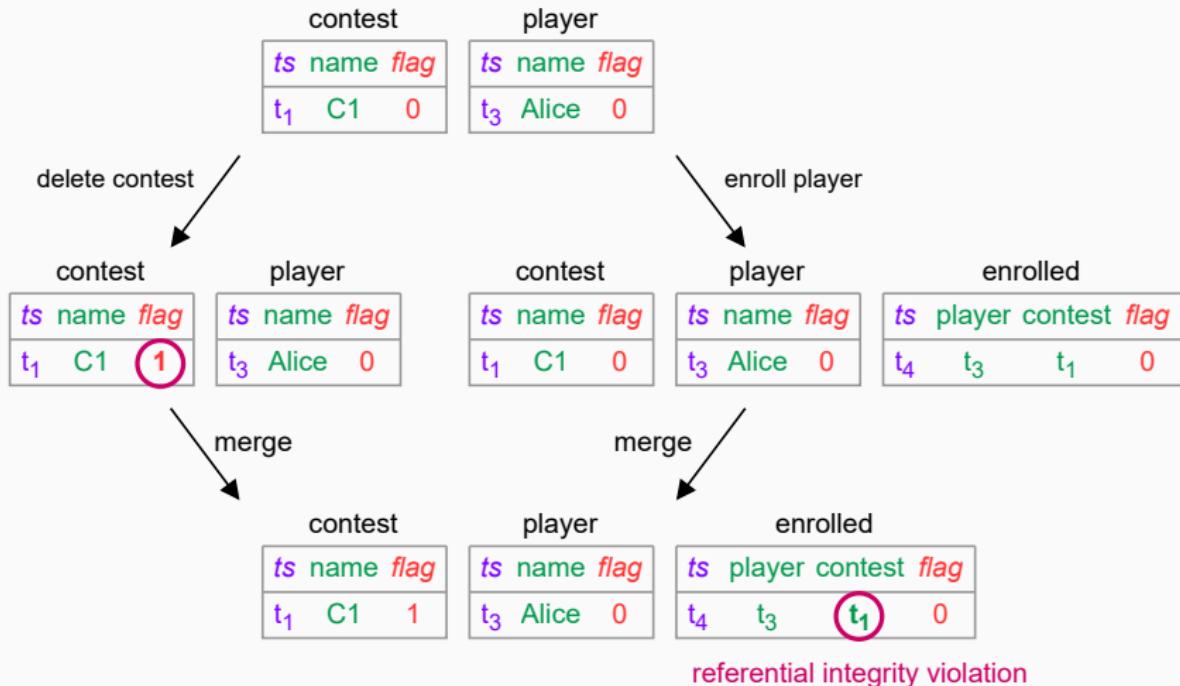
a register for each replicated attribute of r

The diagram illustrates the definition of ReplState as a partial function from relations to tuple identifiers (Ts). It shows that for each relation, there is a register for each replicated attribute and a CLFlag indicating if the tuple is deleted.

app state	contest	player	enrolled																				
	<table border="1"><tr><td>name</td></tr><tr><td>C1</td></tr></table>	name	C1	<table border="1"><tr><td>id_{auto}</td><td>name</td></tr><tr><td>1</td><td>Alice</td></tr></table>	id _{auto}	name	1	Alice	<table border="1"><tr><td>player</td><td>contest</td></tr><tr><td>1</td><td>C1</td></tr></table>	player	contest	1	C1										
name																							
C1																							
id _{auto}	name																						
1	Alice																						
player	contest																						
1	C1																						
replicated state	contest	player	enrolled																				
	<table border="1"><tr><td>ts</td><td>name</td><td>flag</td></tr><tr><td>t₁</td><td>C1</td><td>0</td></tr></table>	ts	name	flag	t ₁	C1	0	<table border="1"><tr><td>ts</td><td>name</td><td>flag</td></tr><tr><td>t₃</td><td>Alice</td><td>0</td></tr></table>	ts	name	flag	t ₃	Alice	0	<table border="1"><tr><td>ts</td><td>player</td><td>contest</td><td>flag</td></tr><tr><td>t₄</td><td>t₃</td><td>t₁</td><td>0</td></tr></table>	ts	player	contest	flag	t ₄	t ₃	t ₁	0
ts	name	flag																					
t ₁	C1	0																					
ts	name	flag																					
t ₃	Alice	0																					
ts	player	contest	flag																				
t ₄	t ₃	t ₁	0																				

The diagram shows the mapping from the app state to the replicated state. Arrows point from each app state entry to its corresponding replicated state entry, illustrating how attributes and flags are replicated across the two states.

Composing CRDTs is not sufficient



Computing a state that ensures data integrity

contest			player			enrolled			
ts	name	flag	ts	name	flag	ts	player	contest	flag
t ₁	C1	1	t ₃	Alice	0	t ₄	t ₃	t ₁	0

compute state
assuming ON DELETE CASCADE

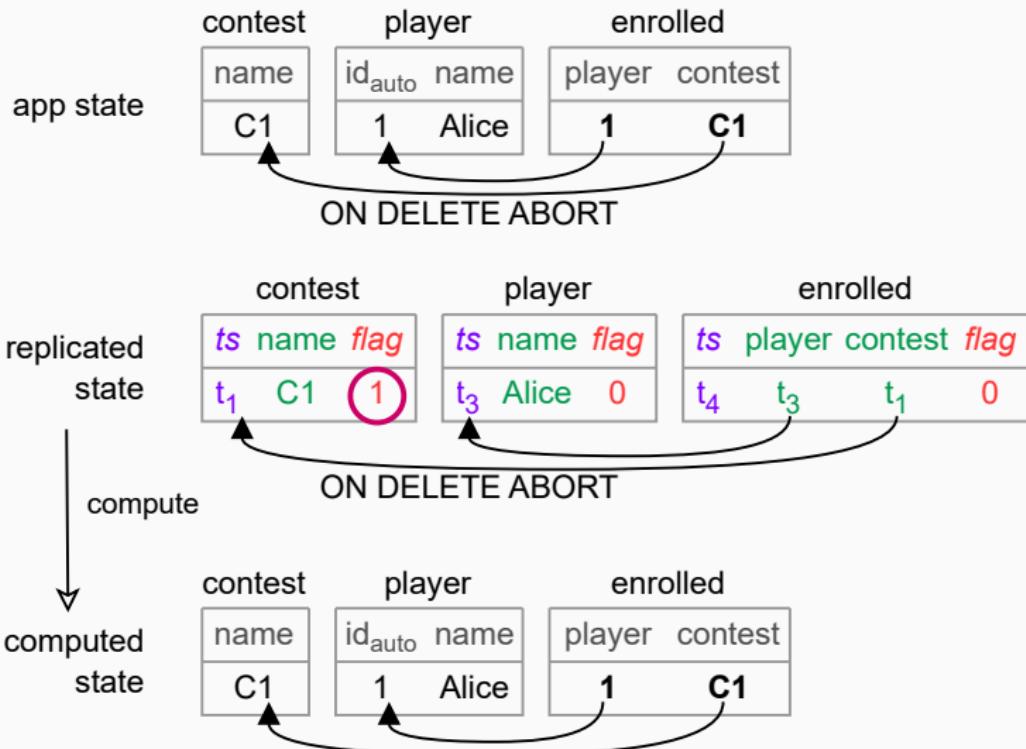
compute state
assuming ON DELETE ABORT

player	
id _{auto}	name
1	Alice

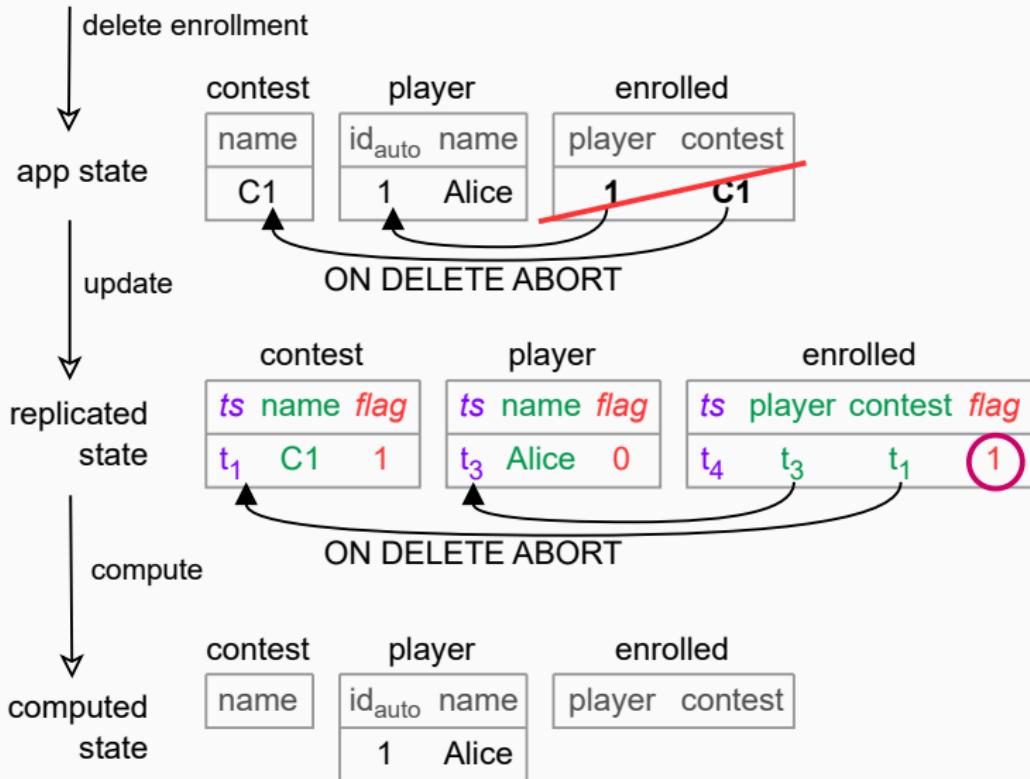
contest		player		enrolled	
name		id _{auto}	name	player	contest
C1		1	Alice	1	C1

- derived from the replicated state
- leverage database schema for selecting computation semantic

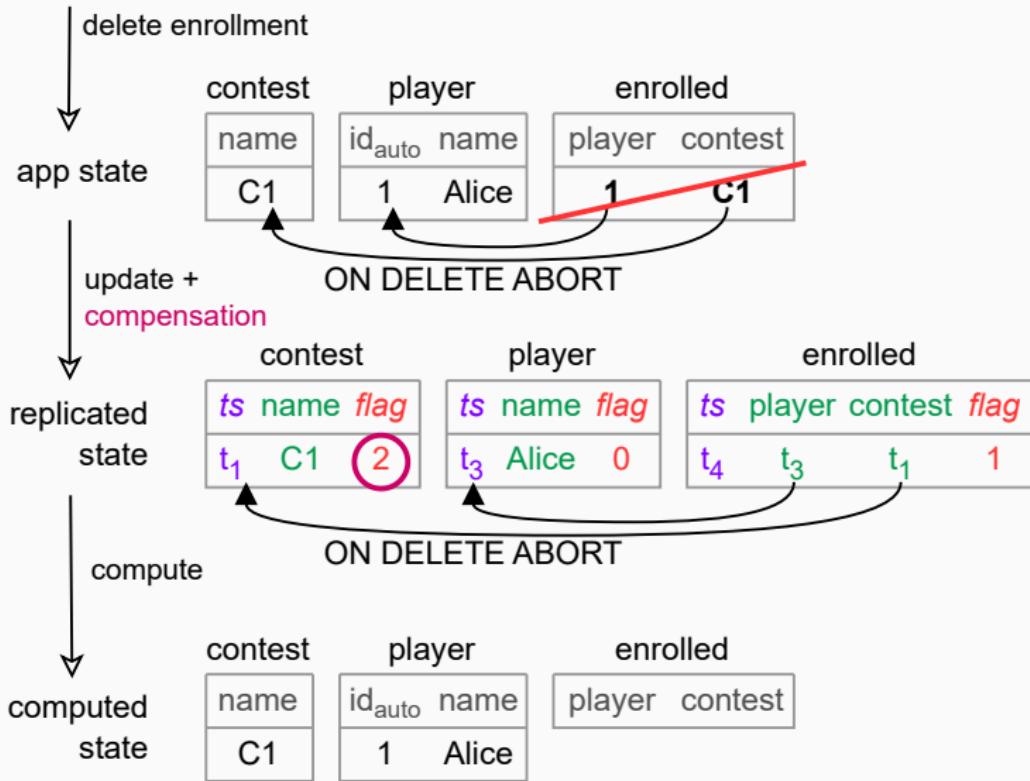
What you see is what you compute



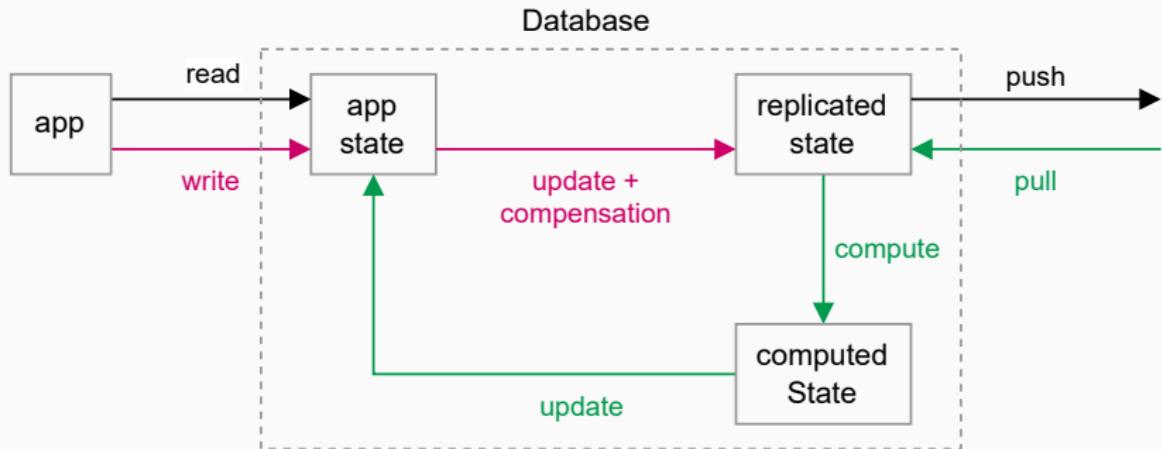
What you see is what you compute



What you see is what you compute



Conclusions



- coordination-less replication of relational database
 - maintains data integrity
 - Strongly Convergent
- composition of CRDTs + state computation + compensations



Victorien Elvinger 4.0