EVENT SOURCING

AND WHEN YOU SHOULD*n't*

- Who has *heard* of event sourcing
- Who has *applied* event sourcing?

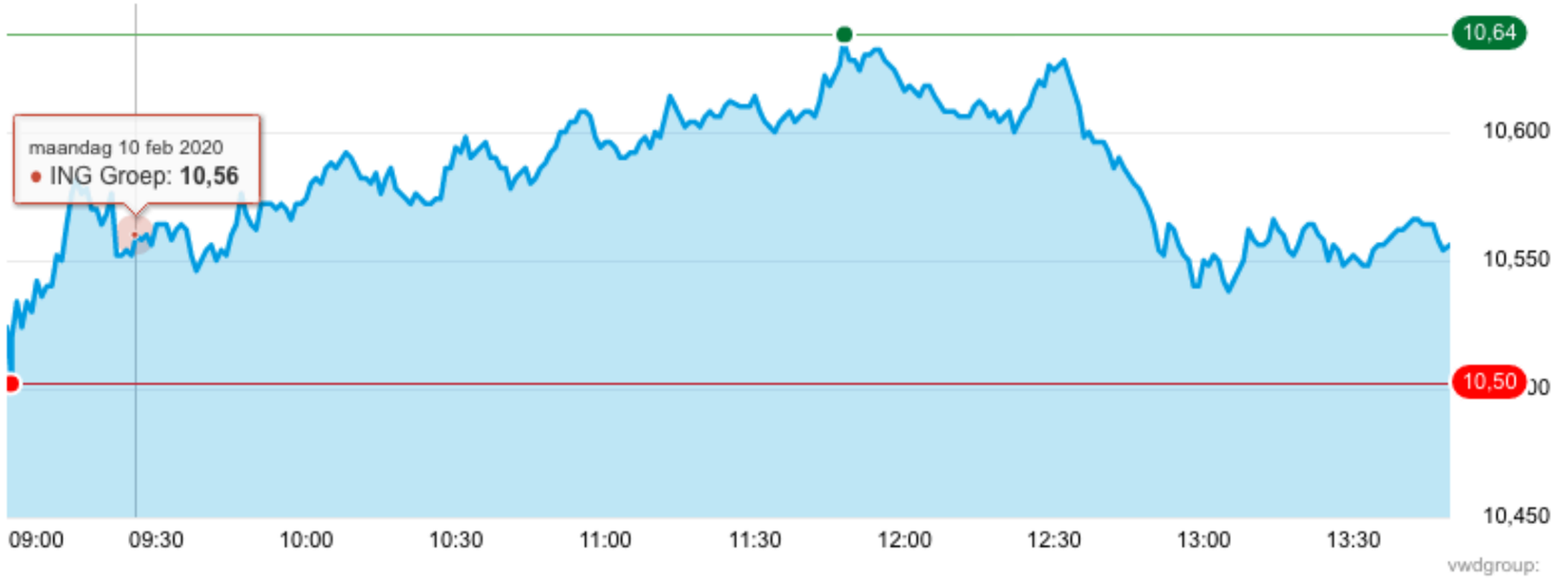
- Creating software, requires a *model* of *reality*.
- A *model* requires a trade-off, we represent a *simplified* reality.
- Usefulness of a *model* is specific to a particular *scenario*


SWEDEN
FINLAND
RUSSIA
POLAND
BELARUS
UKRAINE
FRANCE
ROMANIA
BULGARIA
TURKEY
CYPRUS
LEBANON
SYRIA
IRAQ
KAZAKSTAN
KIRGIZ STEPPE
UZBEKISTAN
TURKMENISTAN
AFGHANISTAN
PAKISTAN
IRAN
TIBET
INDIA
NEPAL
SAUDI ARABIA
OMAN
YEMEN
UNITED ARAB EMIRATES
RUB AL KHALI (Desert)
Tropic of Cancer
ARABIAN SEA
GULF OF ADEN
ETHIOPIA
SOMALIA
DJIBOUTI
SUDAN
EGYPT
LIBYA
ALGERIA
TUNISIA
NIGER
CHAD
CAMEROON
CENTRAL AFR. REPUBLIC
ZAIRE
CONGO
UGANDA
KENYA
TANZANIA
ANGOLA
ZAMBIA
MALAWI
MOZAMBIQUE
SEYCHELLES
COMOROS
MADAGASCAR
MAURITIUS
MALDIVES
SRI LANKA
Direction of the Earth's Rotation
45°
60°
75°
30°
15°N

- How do we model *state* in our application?
- *State* is represented as a *snapshot*.
- But in the real world the *present* is the sum of everything that has happened before
  - A bookshelf is the sum of all the books added to the shelf, and the ones removed.
  - My basket in a store is the sum of all items added to it while walking through the store.

- In many models this *simplification* of reality is fine.
- The same as our *mercator projection*, is *fine* for a map used to travel the seas.
- But what if a *domain* shouldn't be represented as a *snapshot*


Dior



YTD | 1 D | 1 W | 1 M | 3 M | 6 M | 1 Y | 3 Y | 5 Y | ALLES

Lijngrafiek

Interactieve grafiek

	Bied (€)	Laat (€)	
4.951	10,554	10,558	4.363
12.173	10,552	10,56	6.571
11.270	10,55	10,562	6.841
9.503	10,548	10,564	10.229
15.932	10,546	10,566	16.050
12.217	10,544	10,568	12.413
8.871	10,542	10,57	8.821
5.249	10,54	10,572	8.931
4.161	10,538	10,574	6.502
1.776	10,536	10,576	4.396

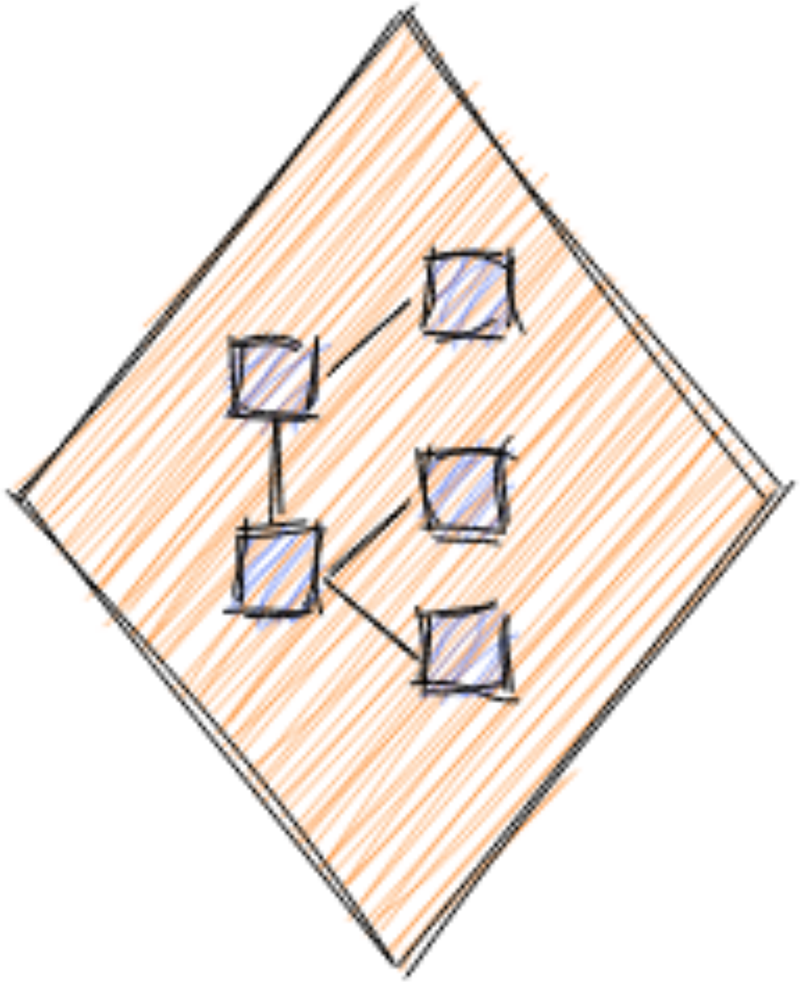

General Ledger
Post. Ref.
Debit
Credit
ON
AC
C

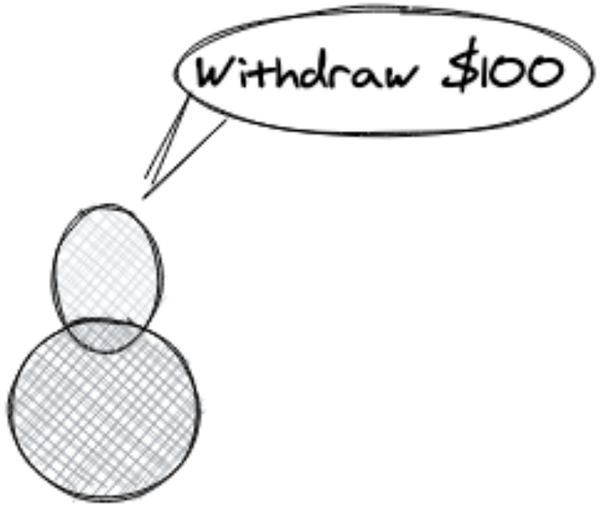
- In a domain where modelling state as a *snapshot* is not a useful *simplification*.
- We can rely on an architectural pattern called *event sourcing*

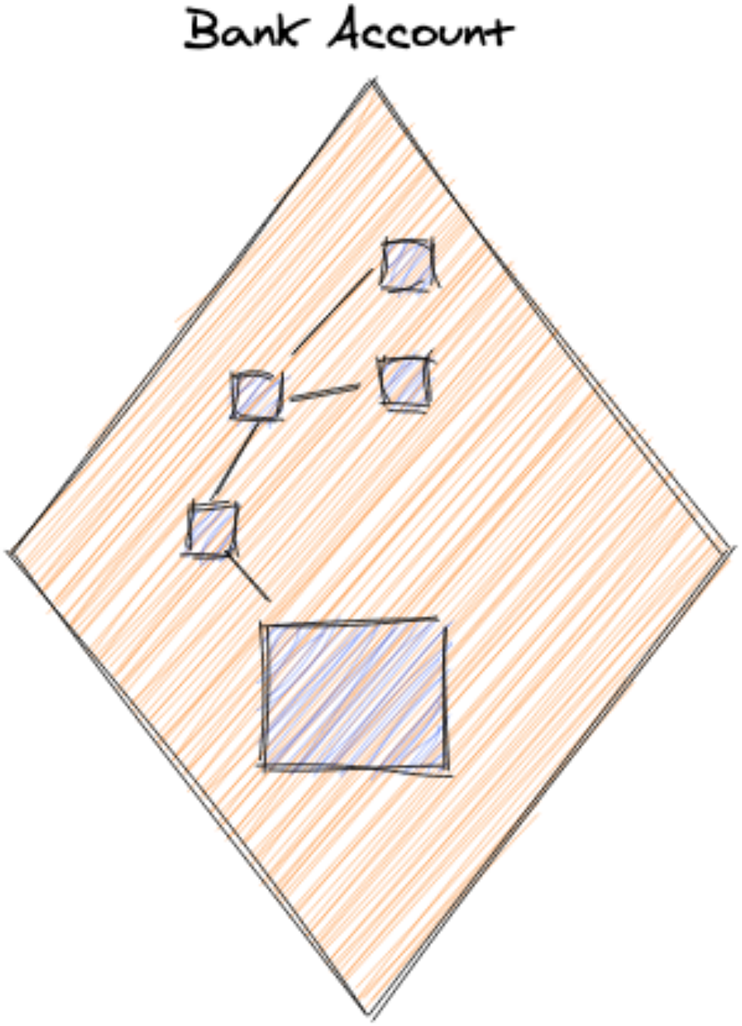
WHAT IS EVENT SOURCING?

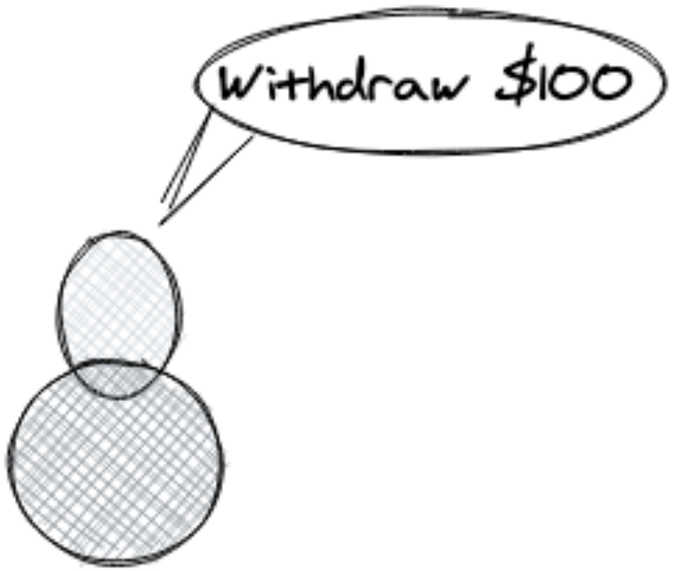
Event sourcing is the architectural pattern where the *full history* of the domain is persisted as a *sequence of facts* (or events). Rather than persisting just the current state.

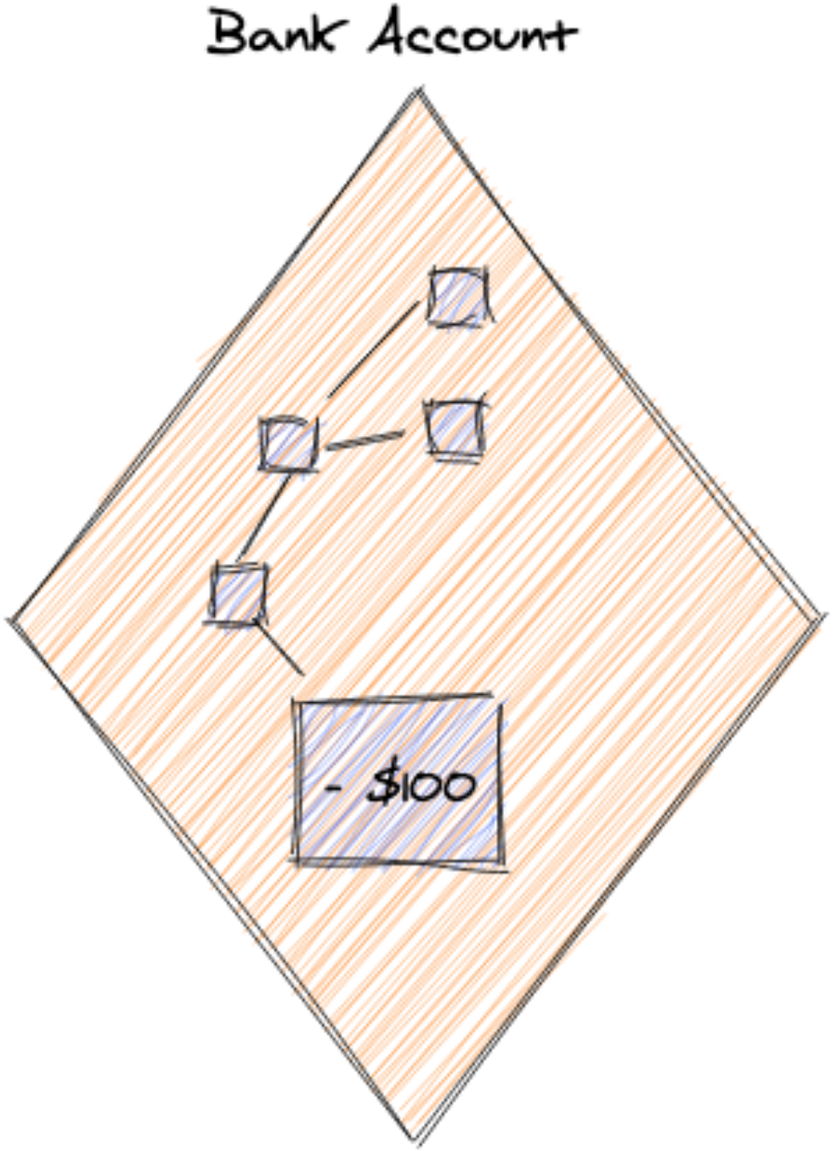
Aggregate Root

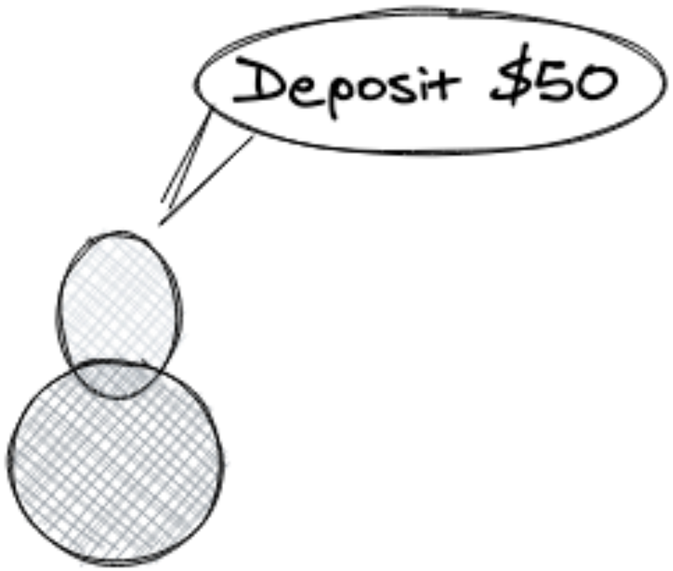


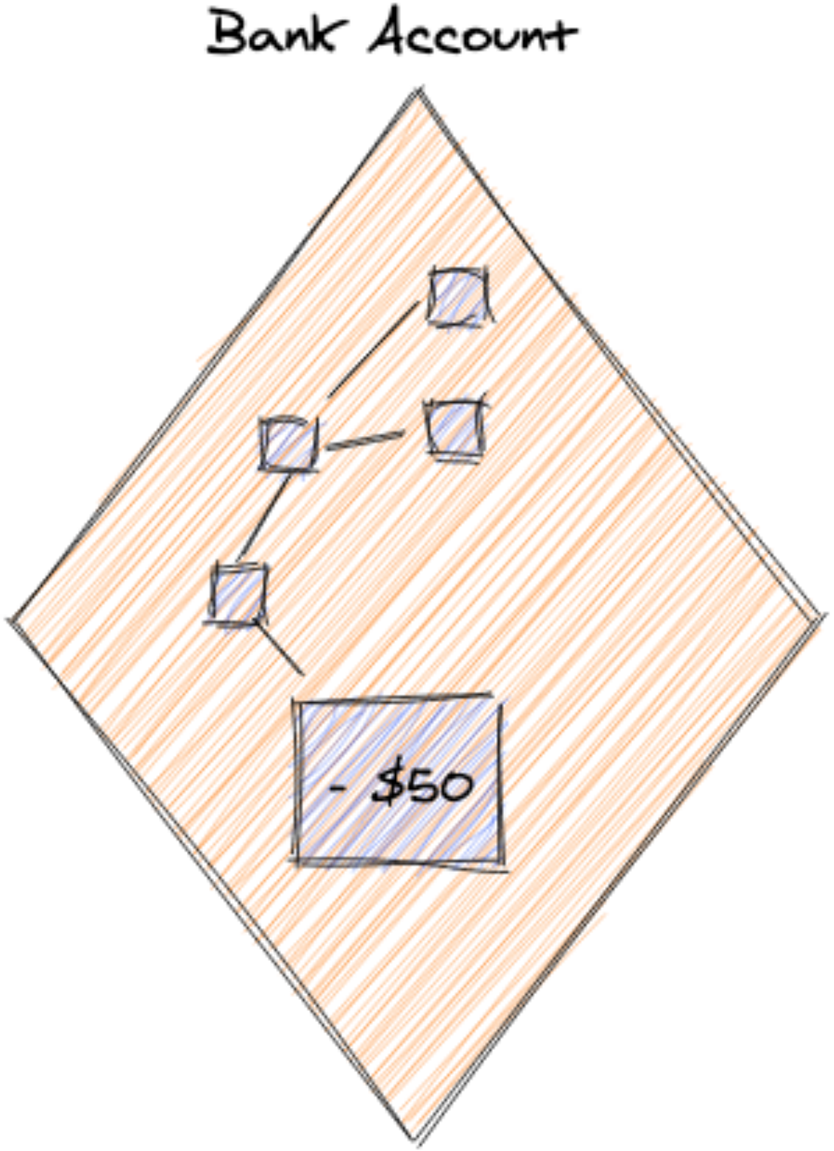

Withdraw $100


Bank Account


Withdraw $100

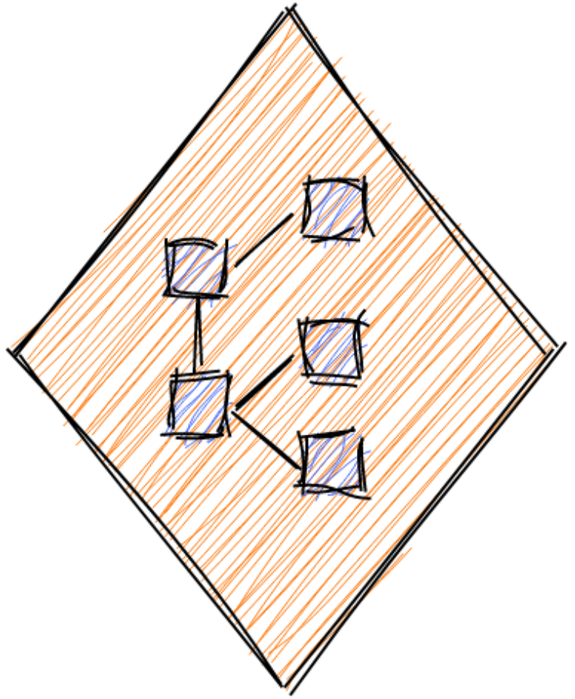

Bank Account
- $100


Deposit $50


Bank Account
- $50

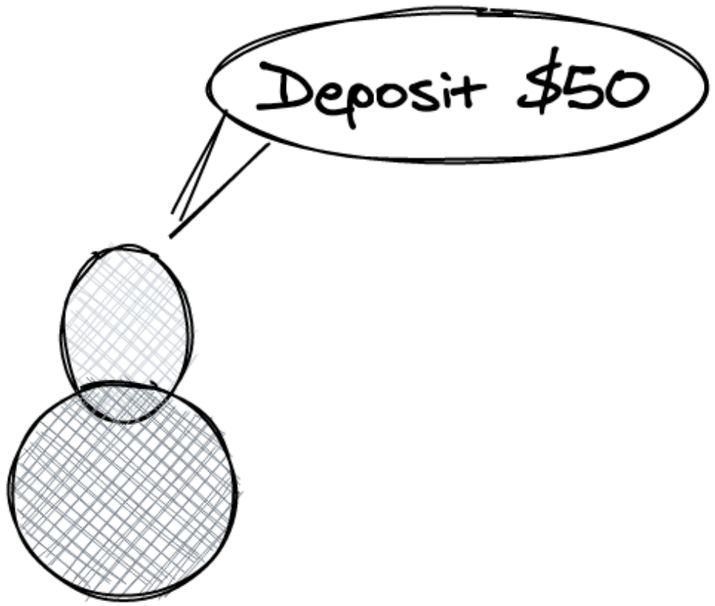
Withdraw $100

Bank Account

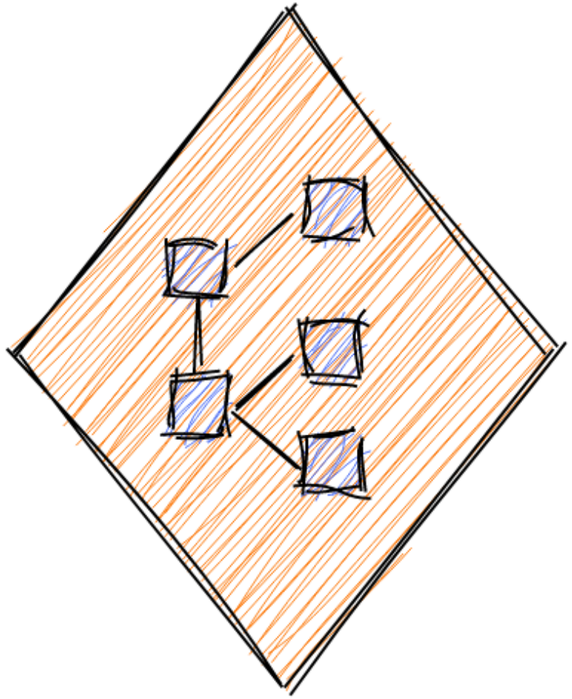


Withdraw

-$100



Bank Account



Withdraw

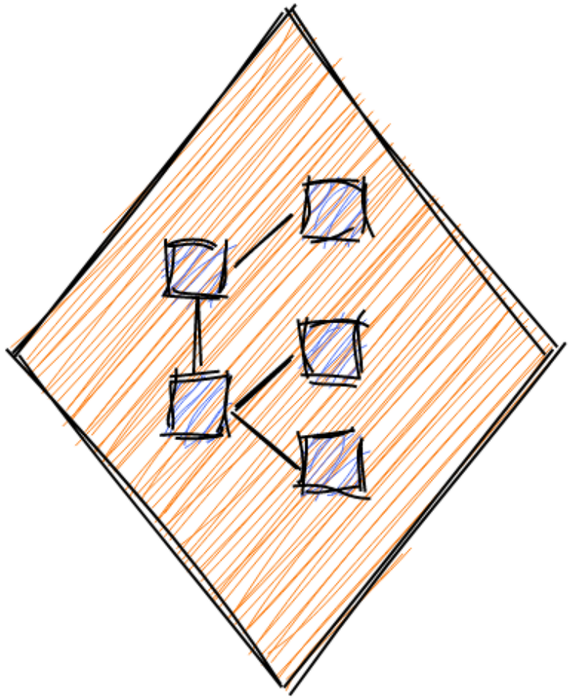
-$100

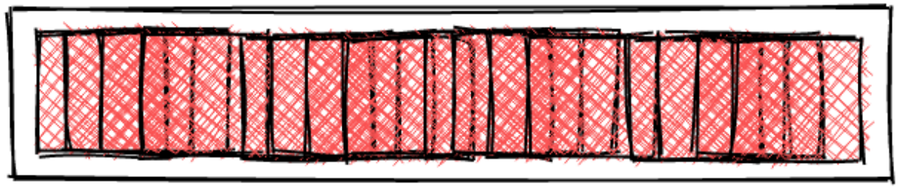
Deposit

+$50

- Audit Trail

Bank Account

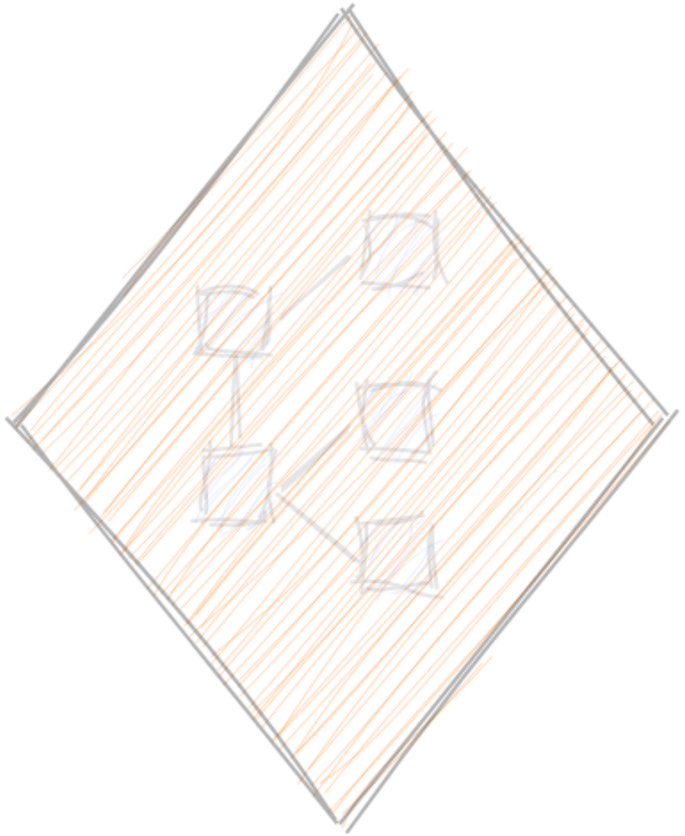


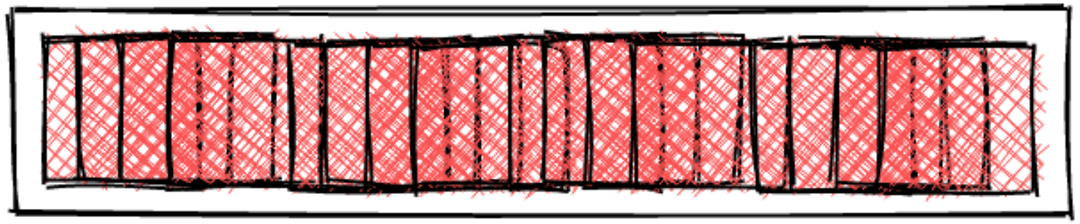


Log

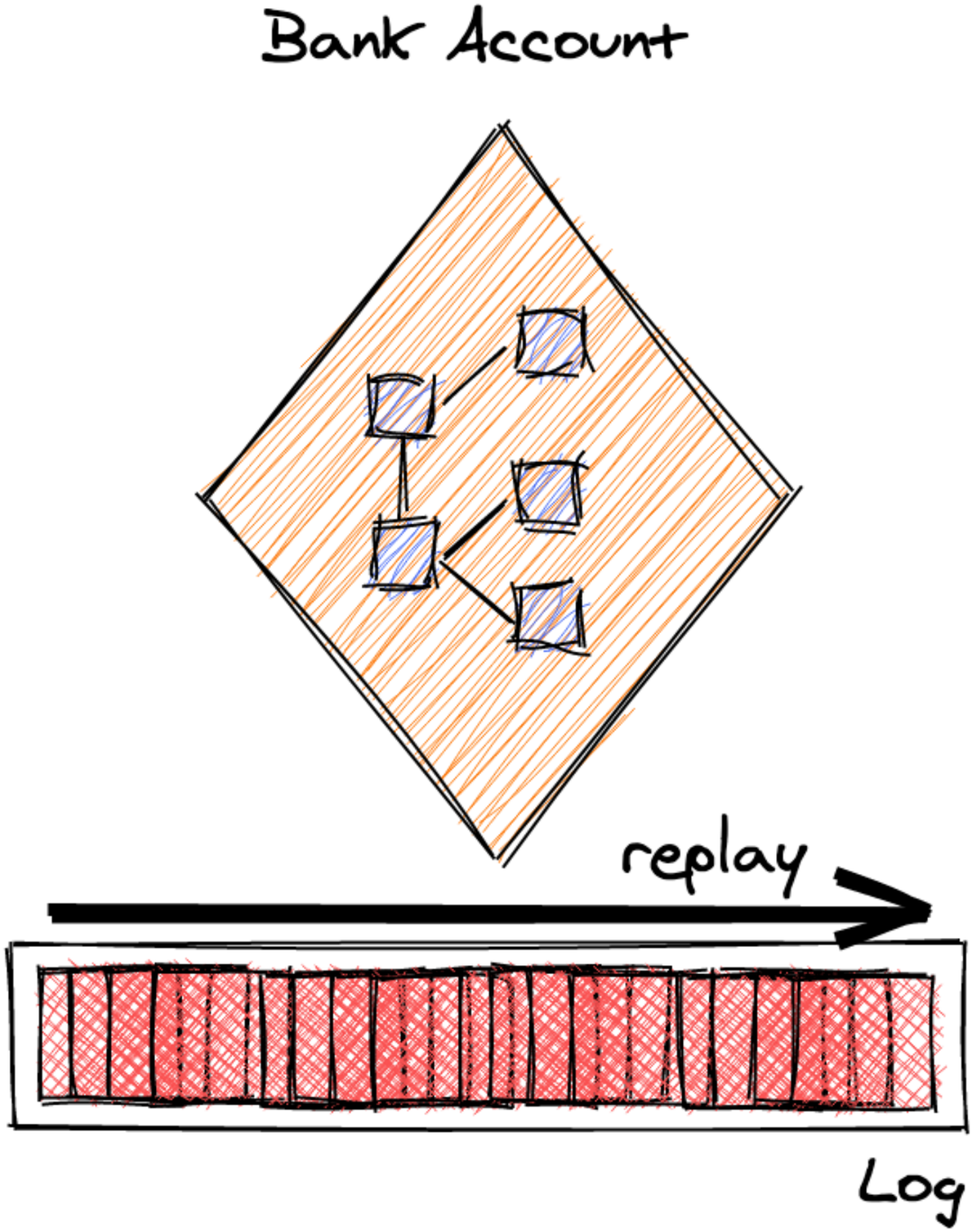
```
var currentState = events.fold(aggregator, emptyState)
```

Bank Account



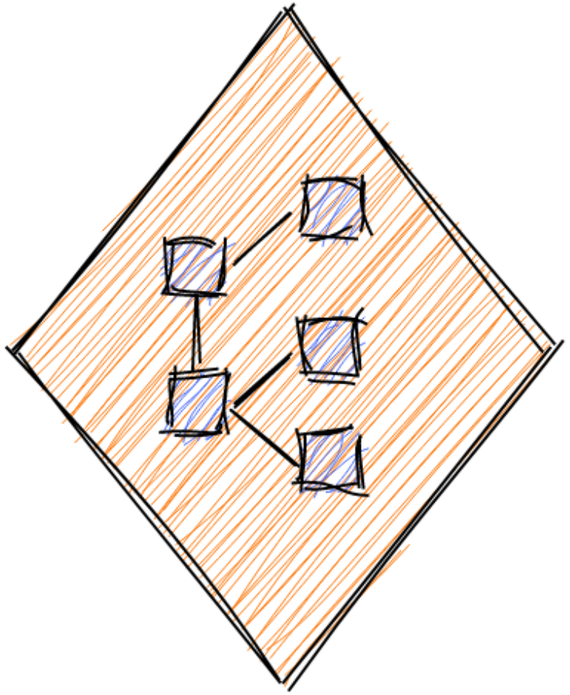


Log

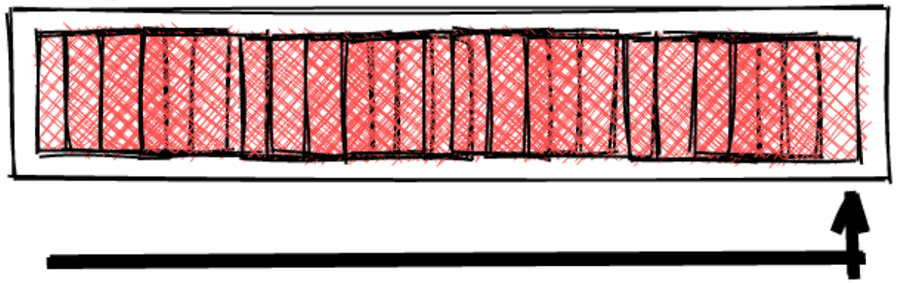

Bank Account
replay
Log

- Audit Trail
- Debugging

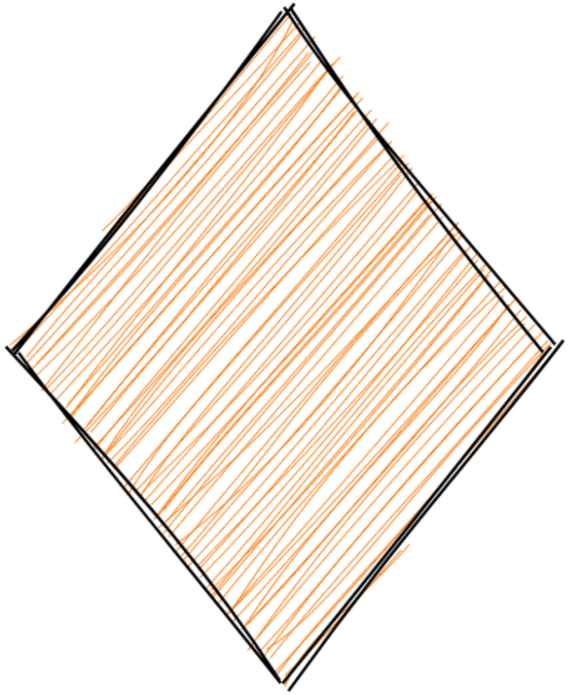
Bank Account



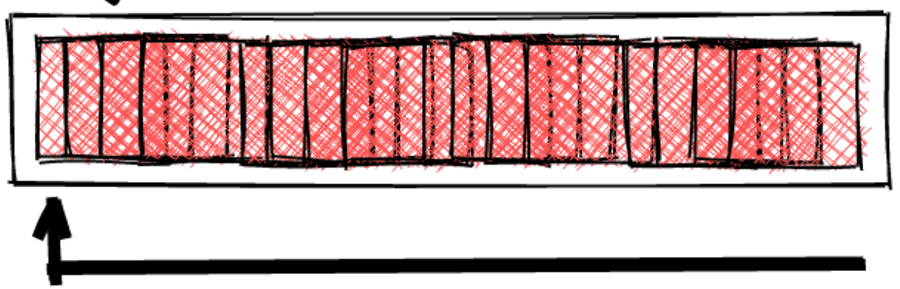
Log



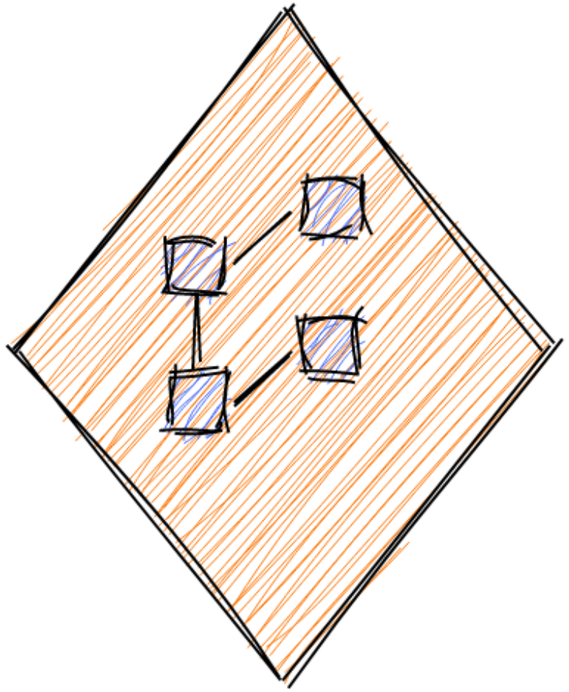
Bank Account



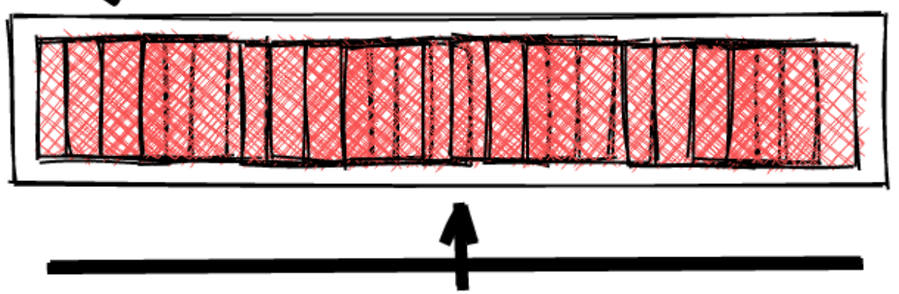
Log



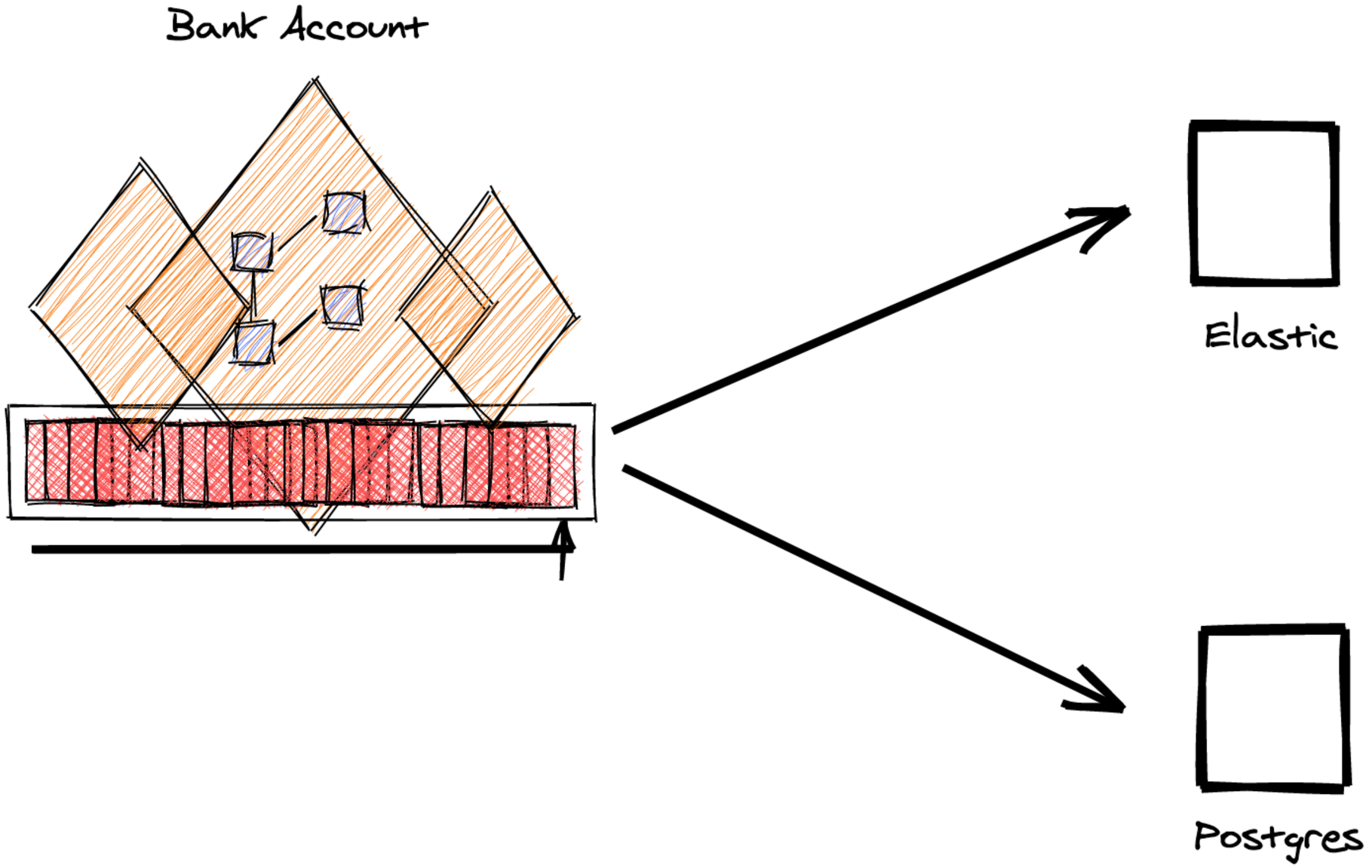
Bank Account



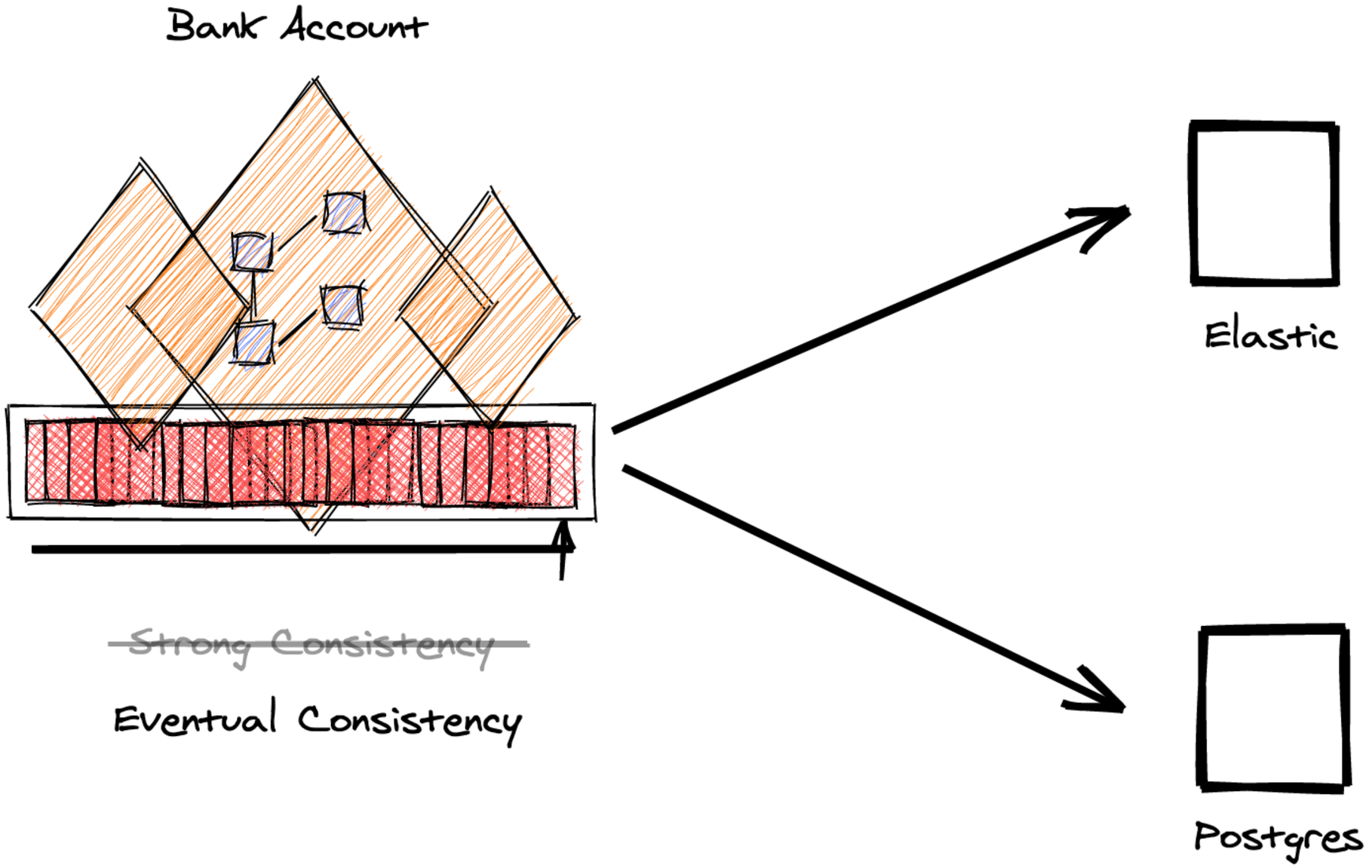
Log



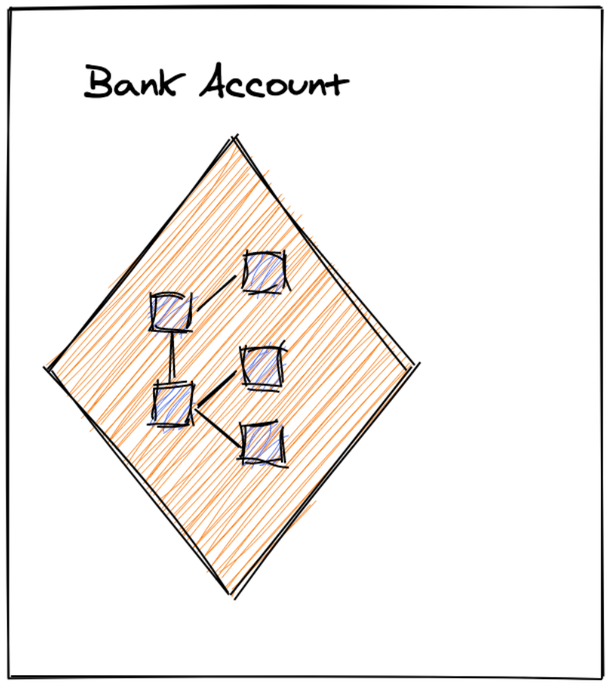
- Audit Trail
- Debugging
- Historic State


Bank Account
Elastic
Postgres

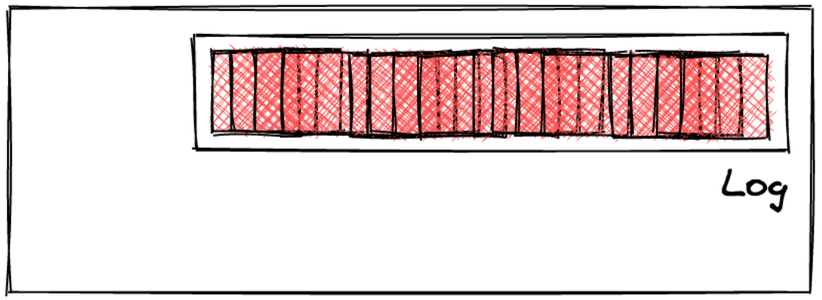
- Audit Trail
- Debugging
- Historic State
- Read models


Bank Account
Elastic
~~Strong Consistency~~
Eventual Consistency
Postgres

In Memory



Persistent

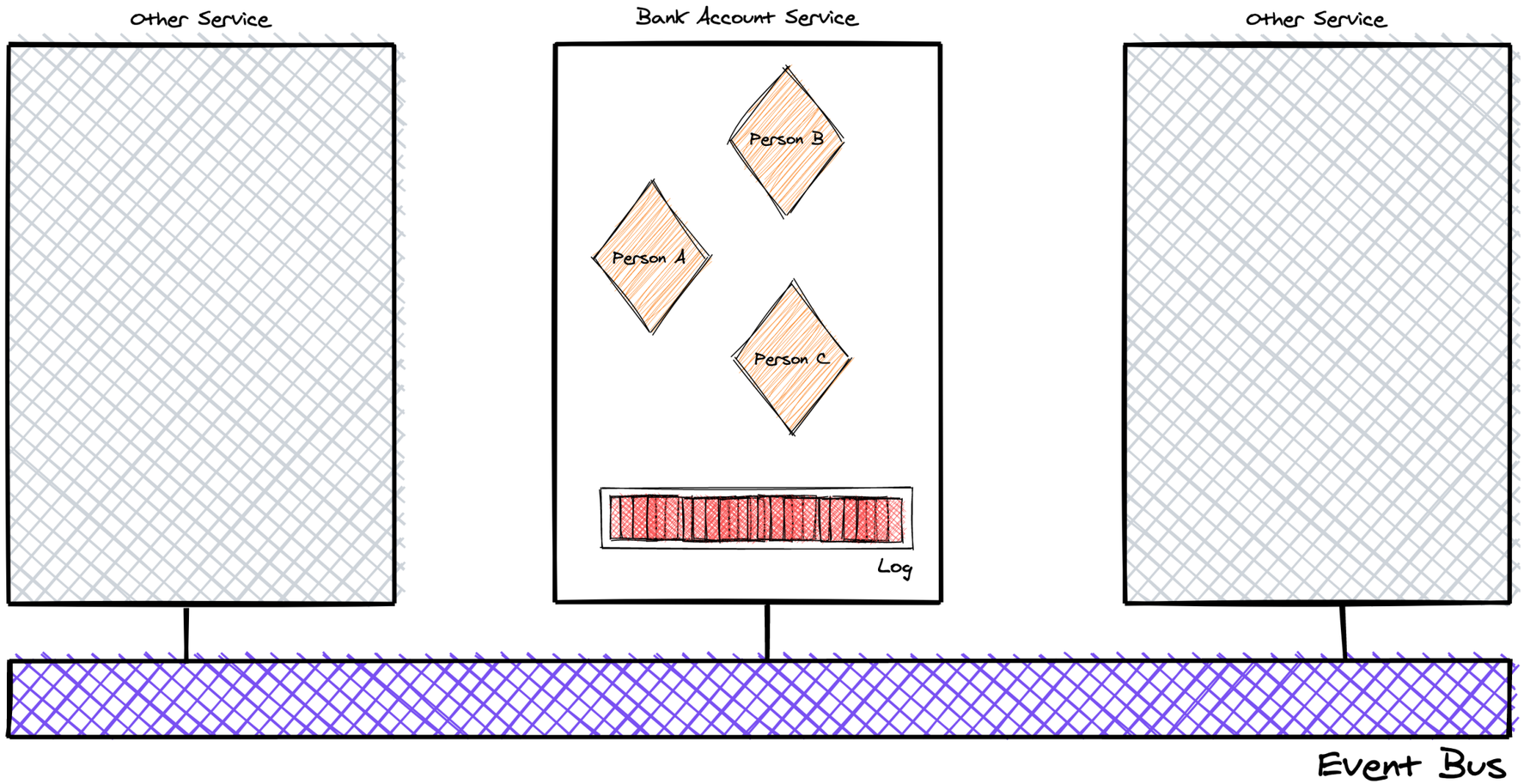


- Audit Trail
- Debugging
- Historic State
- Read models
- In memory processing

COMMON MISTAKES

- Confusing the need for *one* technical capability of the pattern, for needing the pattern.
- Building your own *event sourcing* solution! 👎
- Mistaking event sourcing for *event driven architectures*.

EVENT SOURCING VS. EVENT DRIVEN ARCHITECTURE



COMMON MISTAKES

- Confusing the need for *one* technical capability of the pattern, for needing the pattern.
- Building your own *event sourcing* solution! 👎
- Mistaking event sourcing for *event driven architectures*.
- Event Sourcing is about *business events*, which model the real world. Not *technical events* only relevant to the implementation.

 EVENT SOURCING, WHEN YOU SHOULD*n't*

- We want to apply a *hip* architectural pattern.
- We need an *audit trail*
- For *performance*

EVENT SOURCING, WHEN YOU SHOULD

- When your model needs to represent a *temporal domain*.
- When you are building your own *database*
- When you are building your own *version control system*