



Event Sourcing: Da teoria à prática!

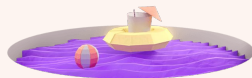
Gophercon Brasil 2022

Andreia Silva

Senior Backend Developer
na Toggl.

- Golang
- Java (oi?)

Meta de vida: Tirar uma foto
com o notebook na piscina.
O notebook eu já tenho.



A Teoria



"We can query an application's state to find out the current **state** of the world, and this answers many questions.

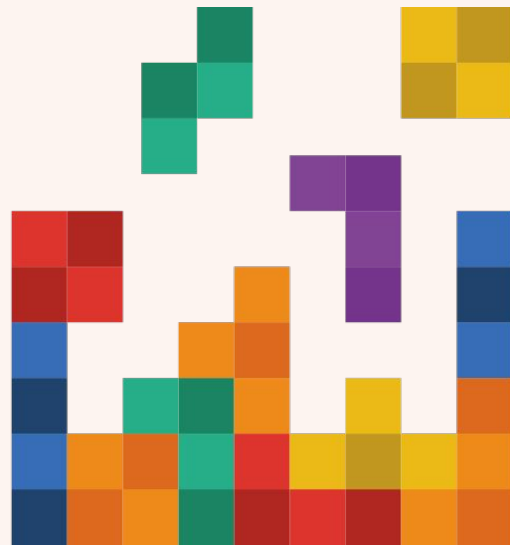
wallet			
id	balance_cents	created_at	updated_at
1	1500	2022-09-29	2022-10-01

However there are times when we don't just want to see **where we are, we also want to know **how** we got there."**

Martin Fowler.

Event Sourcing

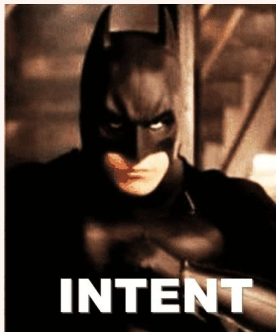
- É um padrão de armazenamento de dados onde o estado da entidade ou agregado não é armazenado.
- A fonte da verdade é representada por uma sequência de eventos que **não deve ser alterada** (*append only*).



Commands vs Events

- Comando (*command*)

- Representa o **verbo**, a intenção de.
- Presente.
- Síncrono.
- Ex: *CreateWallet* ou CriarCarteira.



- Evento (*event*)

- O **fato**. Gerado por algum comando bem sucedido.
- Passado.
- Assíncrono.
- Ex: *WalletCreated* ou CarteiraCriada.

Exemplo

events				
event_id	aggregate_id	version	event_type	event_data
1	1022	1	Created	{ "customer_id": 111, "date": 2022-10-01 ...}
2	1022	2	Credited	{ "amount": 1000 ...}
3	1022	3	Debited	{ "amount": 500 ...}
4	2022	1	Created	{ "customer_id": 222, "date": 2022-10-01 ...}
5	1022	4	Credited	{ "amount": 500 ...}

Como o estado é recuperado?



- *Replay* de Eventos
 - Pode ser um processo custoso.



- *Replay* a partir de *Snapshots*
 - Snapshots são registros periódicos que viabilizam o *Replay* de Eventos a partir de determinado ponto.



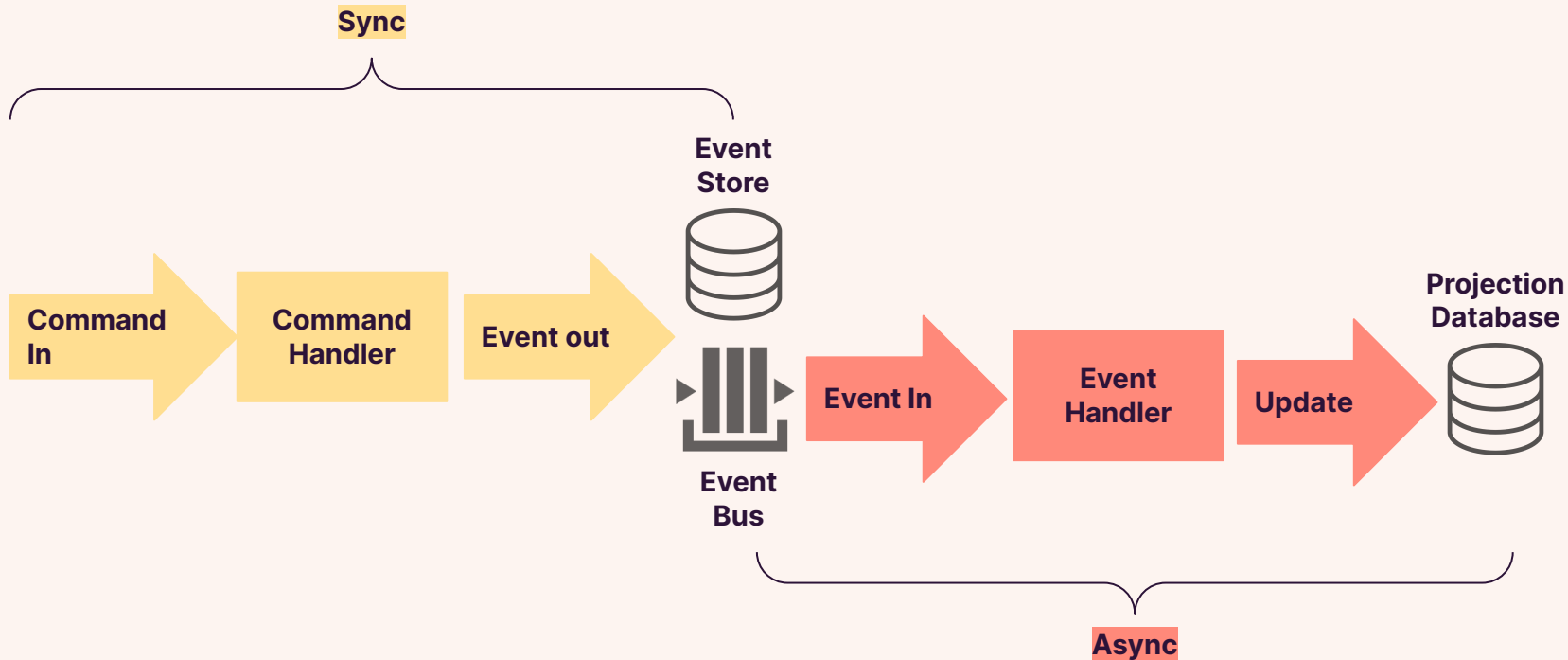
- Projeções (*Projections*)
 - **Visualizações** atualizadas conforme a entrada de novos eventos.

events				
event_id	aggregate_id	version	event_type	event_data
1	1022	1	Created	{ "customer_id": 111, "date": 2022-10-01 ...}
2	1022	2	Credited	{ "amount": 1000 ...}
3	1022	3	Debited	{ "amount": 500 ...}
4	2022	1	Created	{ "customer_id": 222, "date": 2022-10-01 ...}
5	1022	4	Credited	{ "amount": 500 ...}

Balance Projection

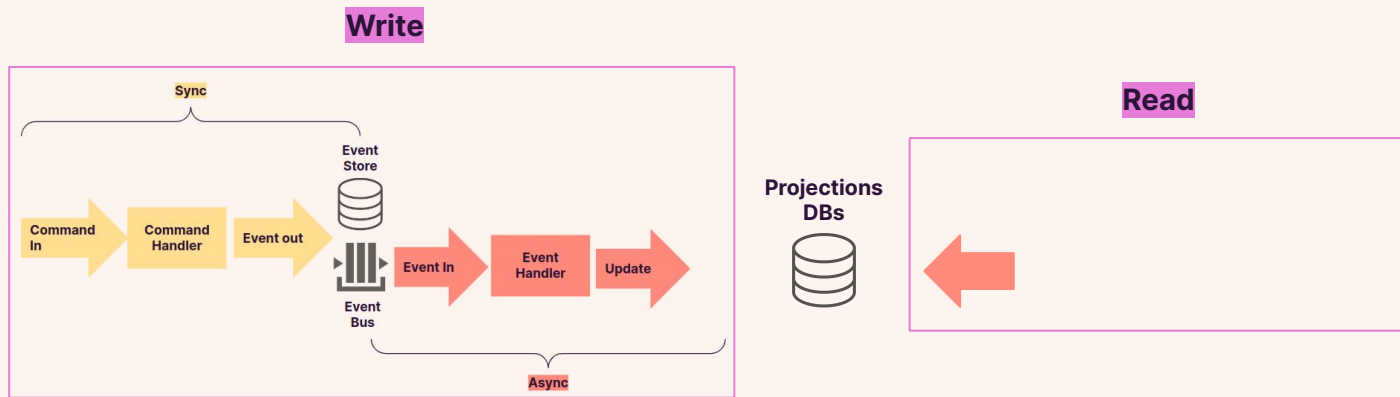
balance		
aggregate_id	version	amount
1022	4	1000
2022	1	0

Fluxo



CQRS

- *Command Query Responsibility Segregation*
 - A utilização de projeções facilita a aplicação de CQRS



Você disse Consistência Eventual?

Peguem as tochas!!!



Quando utilizar Event Sourcing?

Basicamente,

- Quando é importante saber **como** os agregados do sistema atingiram determinado estado
 - Seja por consequência de requisitos regulatórios ou de negócio.
- Possíveis casos de uso:
 - E-commerces
 - Aplicações bancárias / financeiras

Bóra de Event Sourcing então?!

É importante:

- 1) Que o *pattern* “vista” o caso de uso; e
- 2) Pesar prós e contras.



Vantagens

- *Audit Log*
- *Queries* temporais
- *Debug*
- Os dados podem ser utilizados na extração de métricas
- Escrita e leitura podem ser escaladas de maneira independente (CQRS)

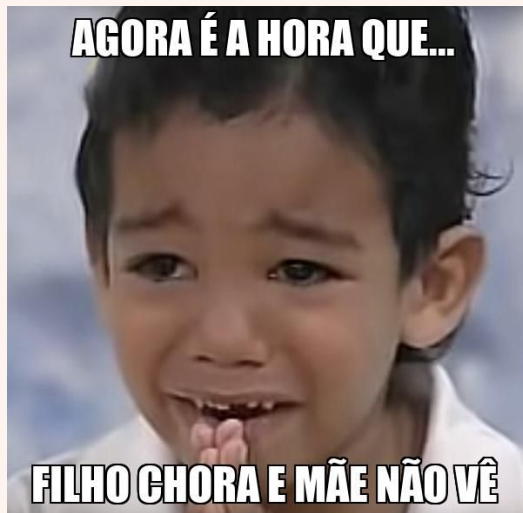


Desvantagens

- Complexidade
- Eventos também representam um histórico de decisões ruins
- Preocupação como tratativa de eventos repetidos, concorrência, design de código...



Agora que conhecemos a teoria, é só traduzi-la para o código.



A Prática



Event Horizon

"[Event Horizon](#) is a CQRS/ES toolkit for Go" (Fonte: GitHub)

- Idealizado por Max Ekman
 - 1.3k estrelas no GitHub
- O Event Horizon reforça padrões e conceitos através de estruturas bem definidas
- Abstrai boa parte da complexidade



Event Horizon

- Por onde iniciar?

Usage

See the example folder for a few examples to get you started.

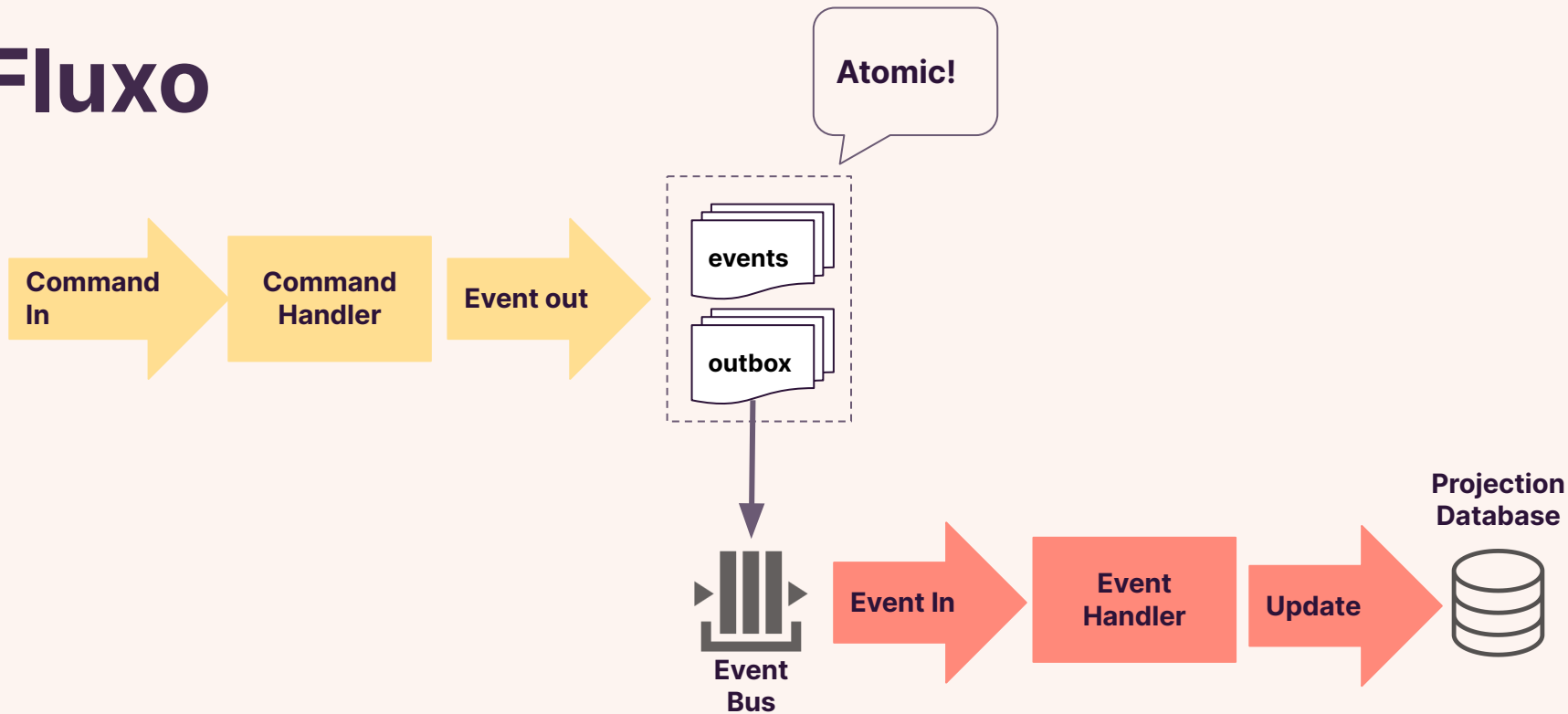
- Atente-se aos Event Stores e Event buses suportados nativamente / *3d party*
- A versão mais recente é v0.15.5.

Event Horizon

- Provê implementação do padrão *Outbox*
 - Evita perda de eventos
 - Auxilia na identificação de gargalos
 - *Track* de eventos
 - *Retry* em caso de erro no processamento de um ou mais eventos
 - * *At least once*



Fluxo



Demo Time!



<https://github.com/andreiac-silva/wallet-api>

Ain, mas esse toolkit tem tudo tão bem definido, isso vai engessar o meu código!



A importância da comunidade



- O código Open Source por vezes é mantido por pessoas como nós, com empregos e vida social.
 - Quando puder, ofereça ajuda!
- Um meetup pode resolver o seu caso de uso
 - Sua experiência pode contribuir muito com outros desenvolvedores

Kudos para Max Ekman e
Rafael Santos.



Obrigada!

 /andreiacamila

 andreiac-silva