

***lunar
way***[®]



**Providing Business Insight
in a Microservice World**



About me

- Started programming on the C64 and the Amiga back in 80s
- M.Sc. in Maths
- Professional software developer since 2004
- Started my career in Java
- Left Java in favour of Scala and FP
- Now primarily working in Go and Typescript

Although I miss Scala's type system, the simplicity of Go is often a bliss



Thomas Bøgh Fangel

Web Architect @ Lunar Way

 [@tbfangel](https://twitter.com/tbfangel)

SIMPLE BANKING
Visa-kort, Konto og App

7-Eleven	40,00
H&M	-9,99
Martin W.	-130,00
Spotify	-18,70
Zalando	-8,50
Uber	
McDonald's	
ASOS	

lunar way

MAKE MONEY MATTER.

lunar way



DIRTY MONEY

lunar way



thelunarway

lunar way

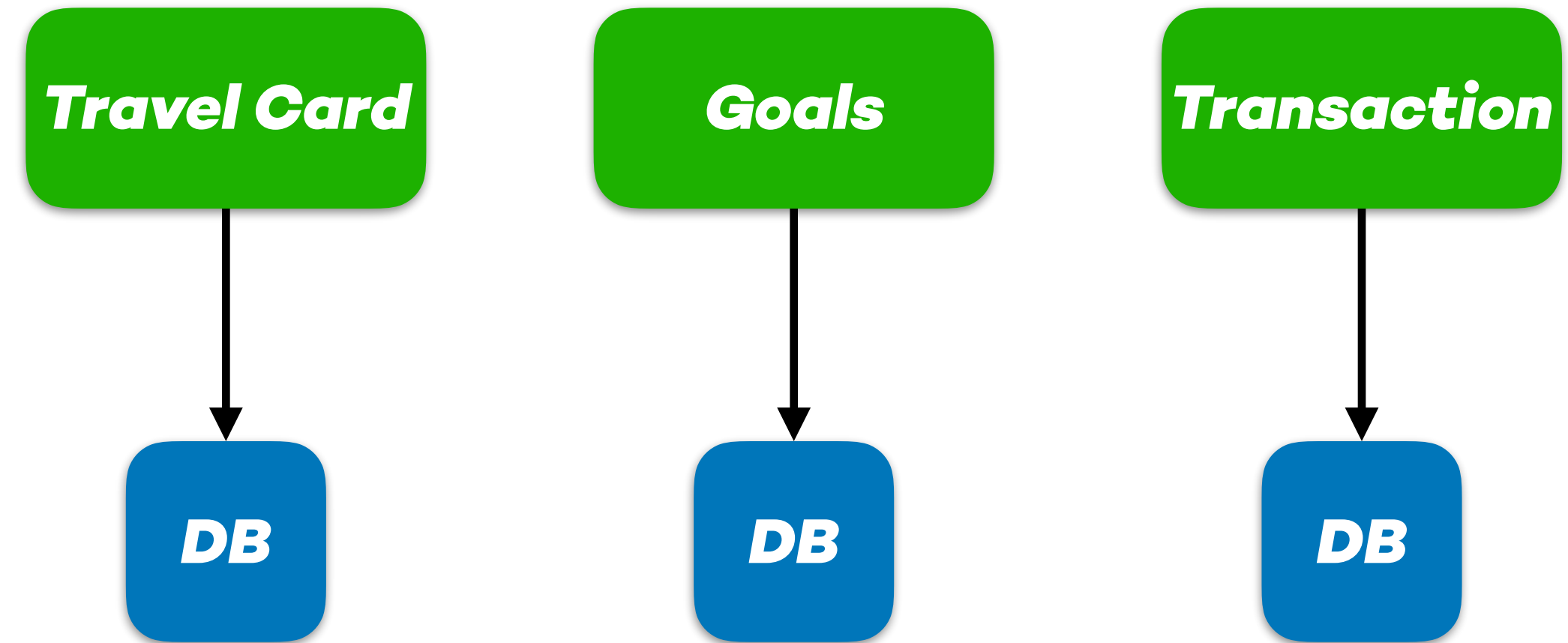
lunar way

KAPSEJLADS

The question

I need to get some numbers on our Travel Card users with at least one travel related goal who have also used the standard Visa card to buy something in a foreign currency!

The problem

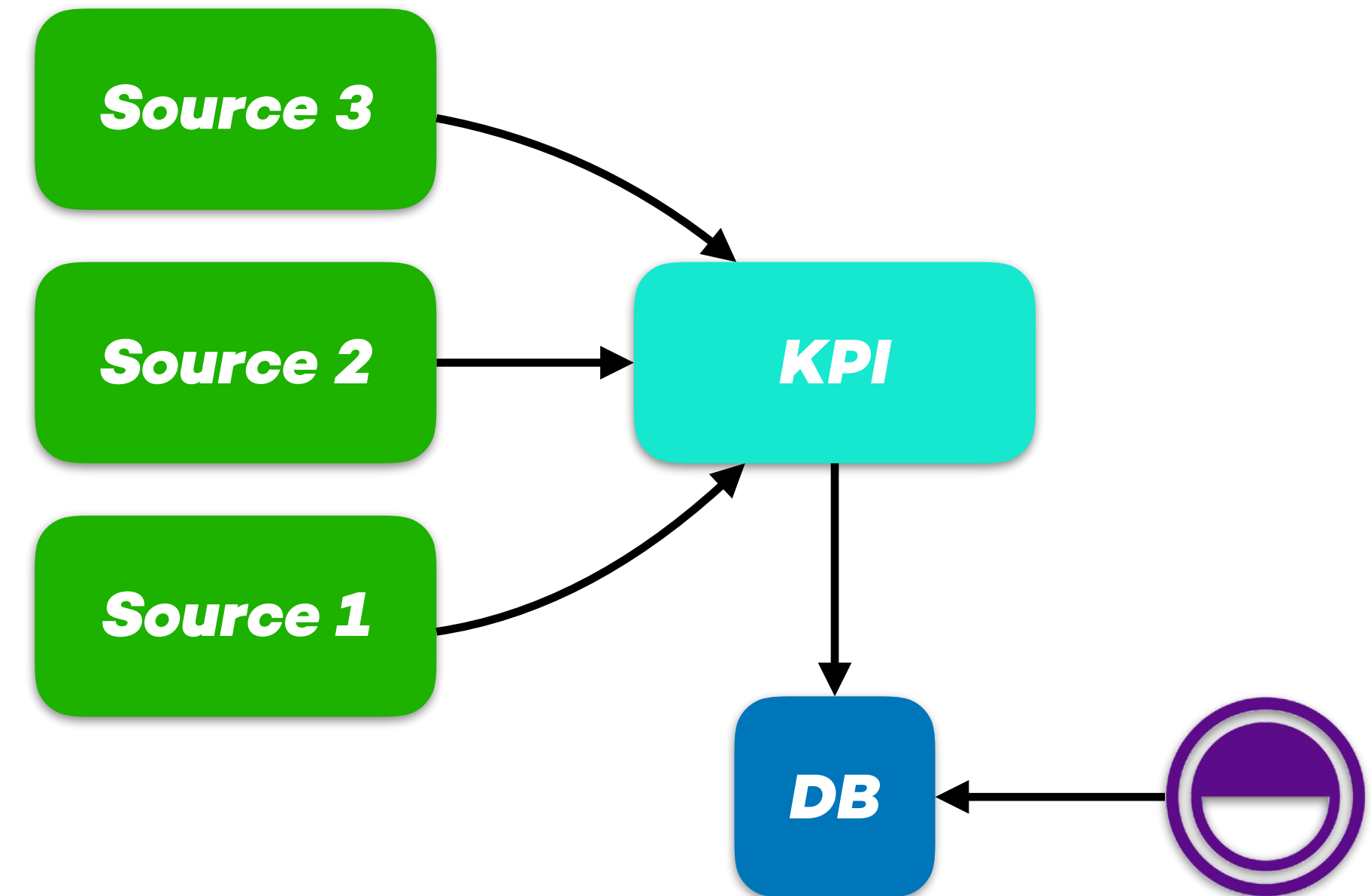


With information spread across several services and databases, how do you provide the required insight?

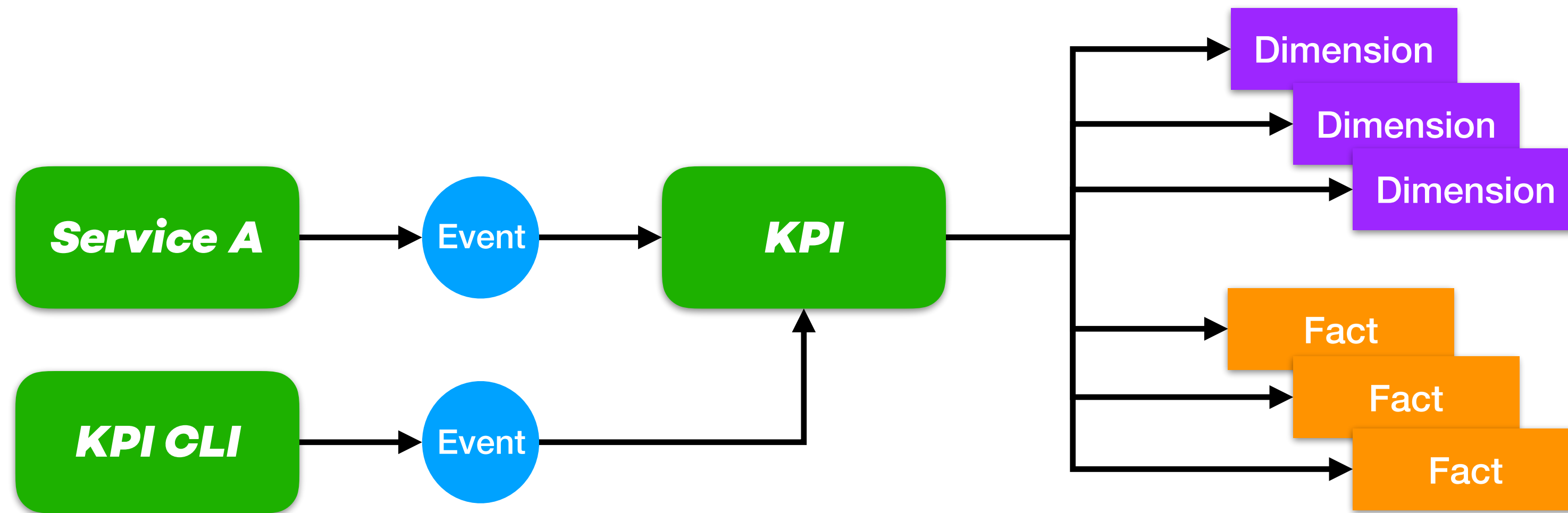
Our solution

Use existing events to build a suitable data model for analytics

- 1. Sources publish events*
- 2. KPI subscribes on events and converts to own model*
- 3. Tooling on top to provide insight*



KPI pipeline



*Sidekick KPI CLI to bootstrap from existing data.
Same pipeline, same events, different queue*

Self service model

Developers must be able to add new KPI pipelines

Feature squads own all aspects of their feature - also KPI

Procedure

- 1. Define facts and new dimensions*
- 2. Generate db schema and repos*
- 3. Implement pipelines*
- 4. Setup event consumers*
- 5. Implement bootstrap*

Fact model and repo

Facts and dimensions are structs
Template based db generation

```
type AccountBalanceChange struct {  
    ID          string  
    UserID      string  
    AccountID   string  
    Created     cdt.TimeD  
    Balance     decimal.Decimal  
    Available   decimal.Decimal  
    Usable      decimal.Decimal  
    Currency    cdg.Currency  
}
```

```
$ make genfactrepo SUBP=account TYPE=AccountBalanceChange  
go run ./cmd/fact_generator.go -force -schema -repo -mockrepo -repos -subp account -name AccountBalanceChange  
Generating file ./pkg/db/fact/account/account_balance_change_schema.go  
Generating from template fact_schema.gotmpl  
Generating file ./pkg/db/fact/account/account_balance_change_repo.go  
Generating from template fact_repo.gotmpl  
Generating file ./pkg/db/fact/account/account_balance_change_mock_repo.go  
Generating from template mock_fact_repo.gotmpl  
Generating file ./pkg/db/fact/account/account_repos.go  
Including type NemKontoActivity in repos init for package account  
Including type AccountBalanceChange in repos init for package account  
Including type Account in repos init for package account  
Including type AccountCreditLimitChange in repos init for package account
```


Pipelines

```
func initAccountBalanceChangePipeline(opts ...pip.PipelineOption) pipeline.Pipeline {  
    options := configureAndValidateAccountOptions(opts...)  
    return pipeline.NewPipe(pip.DefaultPipelineOptions()...).  
        Map(mapFromBalanceUpdatedEvent()).  
        Then(pip.EnsureDimensions(options)).  
        Then(pip.EnsureUser(options)).  
        Then(pip.EnsureAccount(options)).  
        Then(func(entity interface{}) error {  
            fc, err := fact.AsFactContainer(entity)  
            if err != nil {  
                return err  
            }  
            return options.AccountFactRepos.AccountBalanceChangeRepo.InsertFirst(fc)  
        }).  
        End()  
}
```


Event consumers

```
balanceConsumers := consumer.New(account.ExchangeBalance, ConsumerGroupKpi,  
    consumer.WithLogger(zlog.App),  
    consumer.WithRoutingKeyFromEventNaming(account.Events().AccountRoutingKeyCreator),  
    consumer.Pipe(account.Events().BalanceUpdated, accountPipes.AccountBalanceChangePipe),  
)
```


Wrapping up

Key takeaways

If doing microservices you must think about how to deliver answers to cross-service questions

Business model events are very well suited for this kind of added functionality

Autonomous teams should also own a feature's KPI

lunar
way®

Thank you!
Questions?