AKVEO

CQRS and Event Sourcing

in backend systems

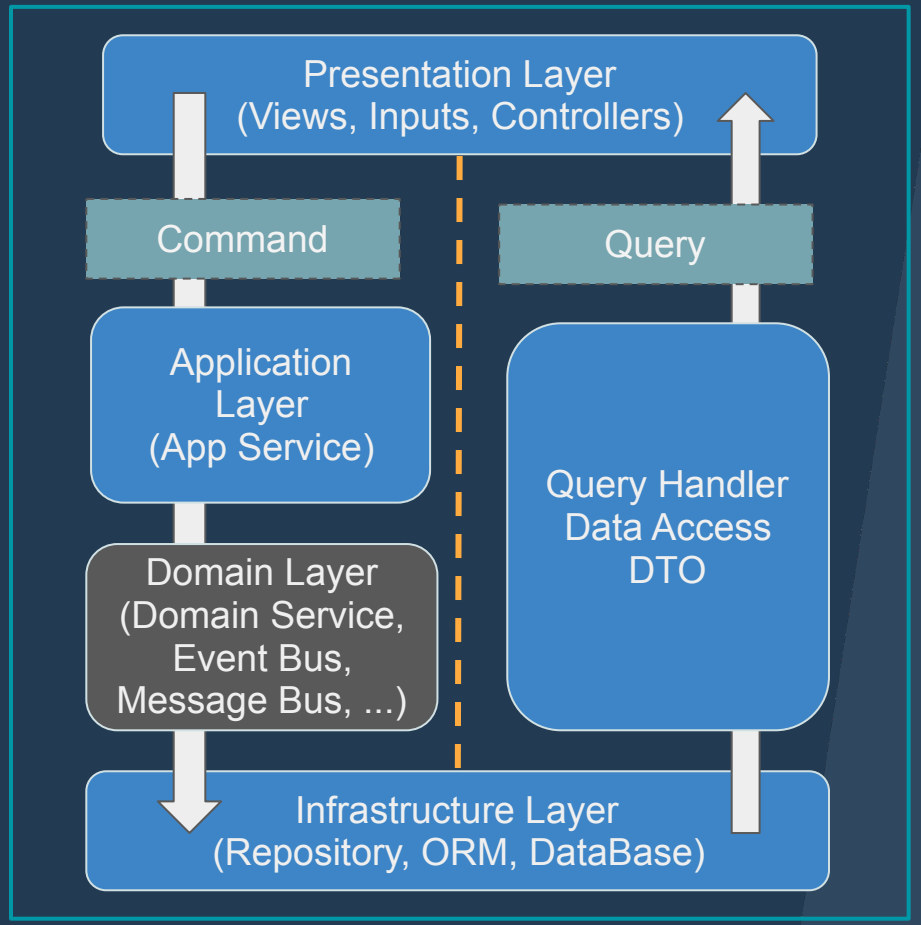
Valentin Kononov


AKVEO

CQRS

CQRS Meaning

- Command
- Query
- Responsibility
- Segregation



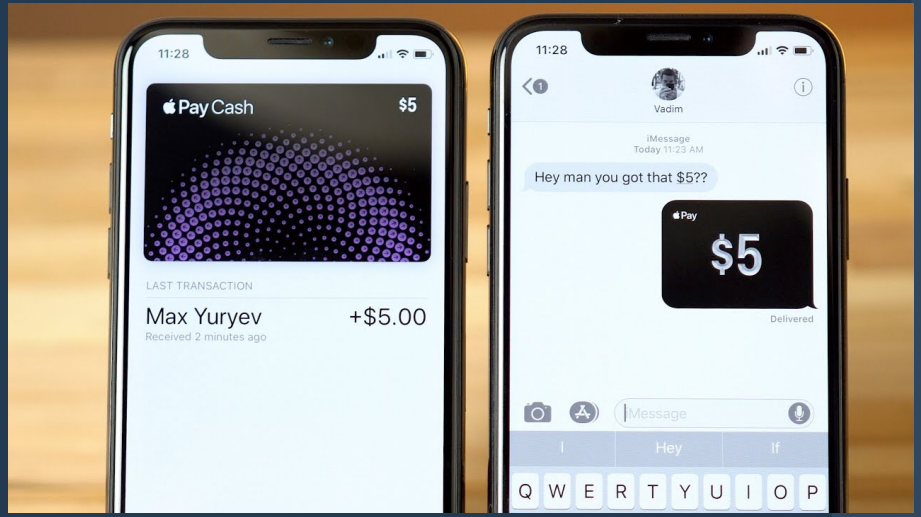
CQRS Concept

Command = workflow, that alters the current state of the system

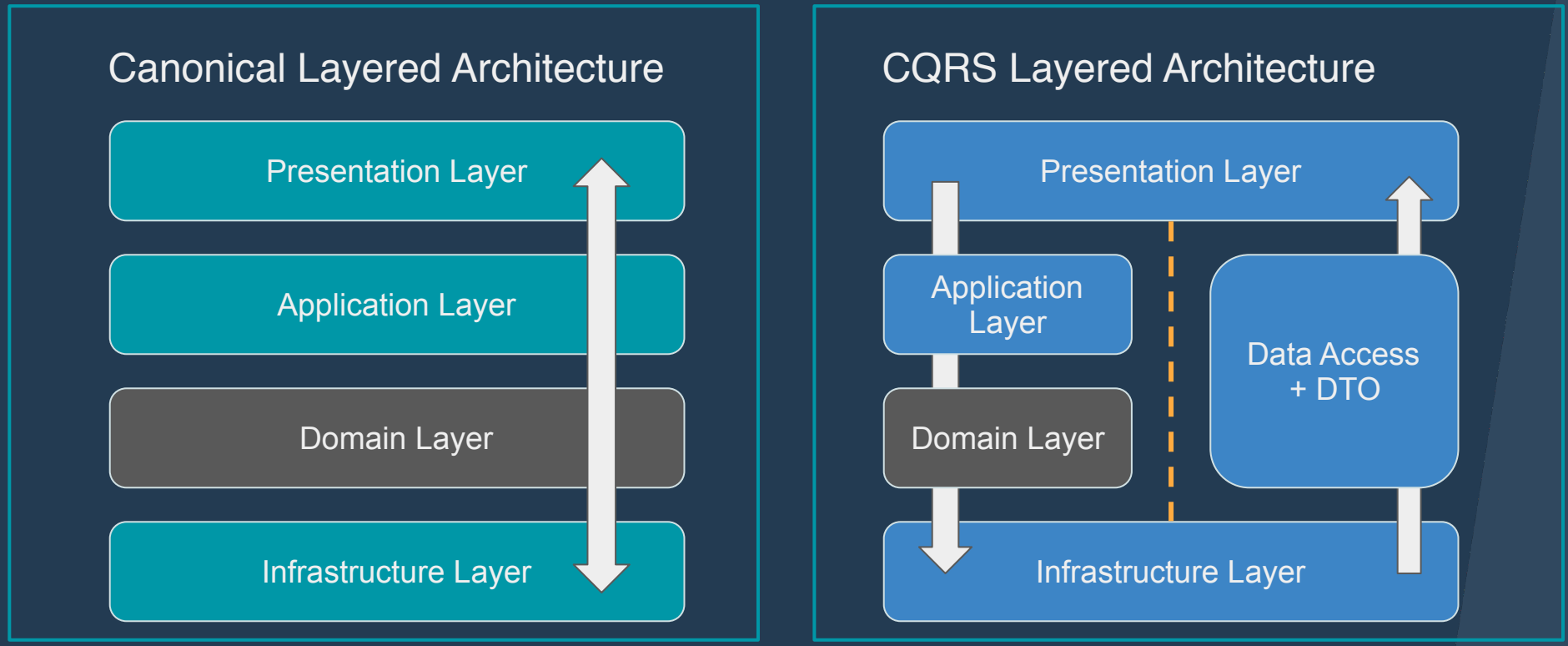
Query = workflow, that reports a view of the current state of the system

Money Transaction Sample

- User performs money transfer
- Most probably he has money
- We can perform transaction and report success
- In case of error - rollback later



CQRS Diagram



CQRS Benefits - Divide et Impera

- Distinct Optimization
- Scalability Potential - both technical and Teams
- Simplified Design and Less complexity of changes

CQRS - In simple code sample

Regular Code

```
private int x;

public int increment_and_return_x()
{
  lock x;
  x = x + 1;
  int x_copy = x;
  unlock x;
  return x_copy;
}
```

CQRS Code

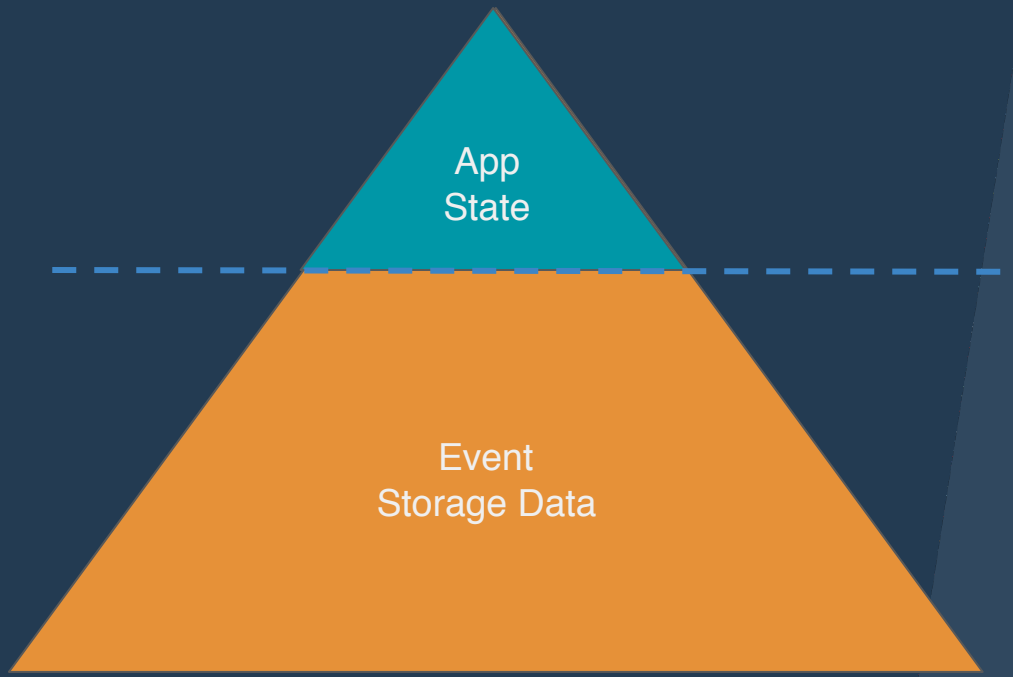
```
private int x;

public int value()
{
  return x;
}

void increment_x()
{
  x = x + 1;
}
```

AKVEO

Event Sourcing

Event Sourcing

- In the real world we observe events
- In software, we tend to write models

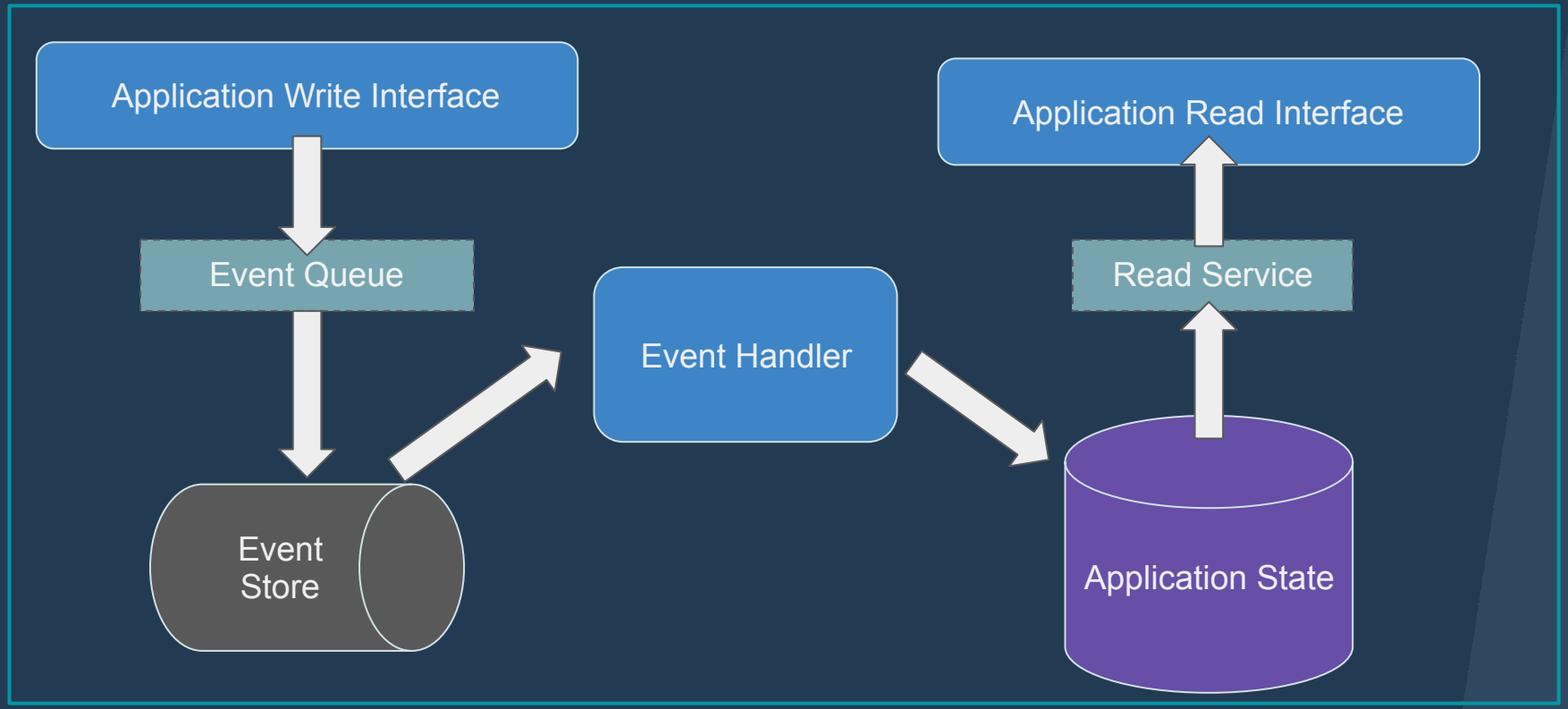


In some cases events are real data

- All Changes to the Application are saved as a sequence of events

- Events are immutable. You don’t miss a thing. For the entire lifetime of the system

Event Sourcing Schema



Read Model

Artificially created data model that offers a business-specific view of the current state of the system.

About Scalability

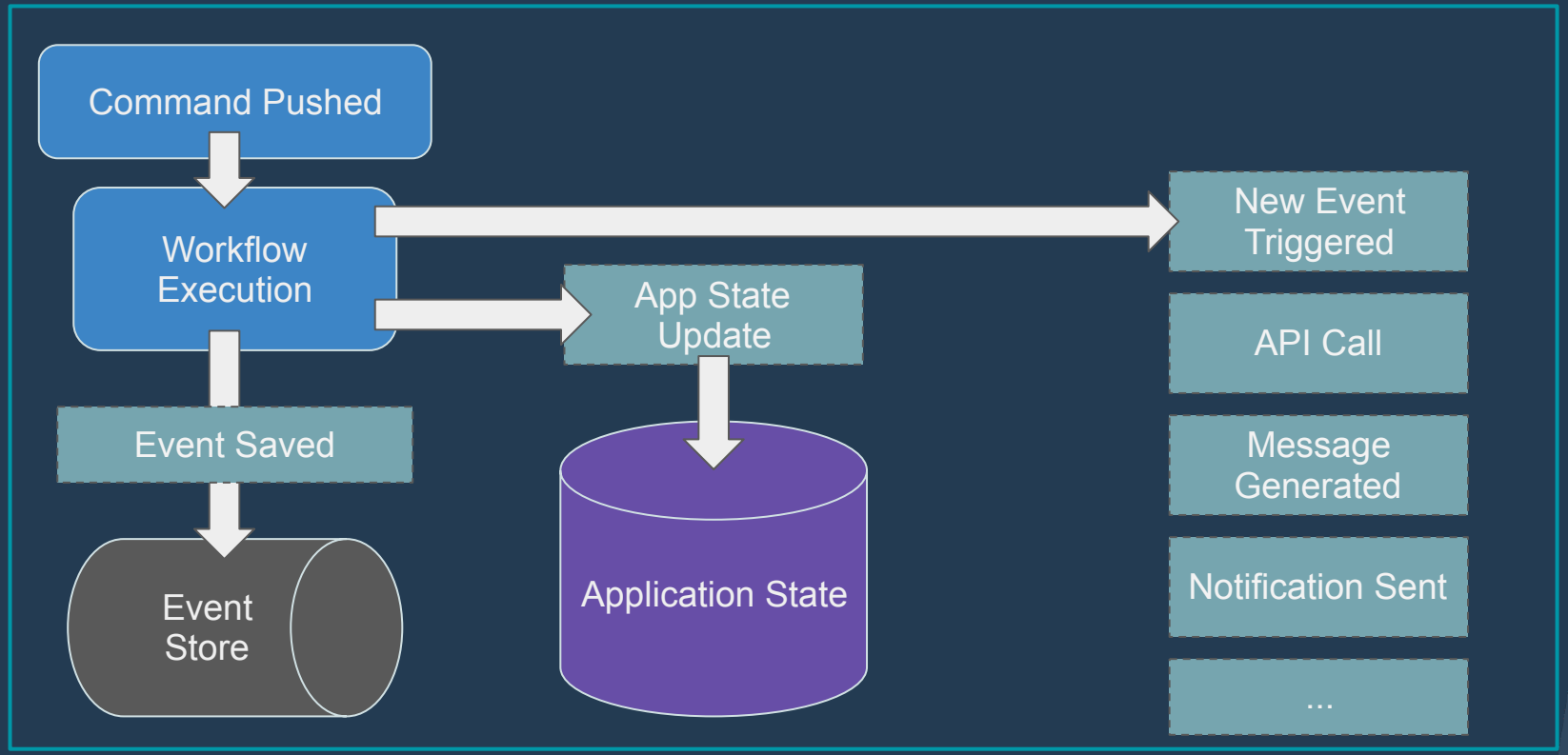
- Event storage allows to implement any amount of new features and services based on the same data
- Consistency - minimal set of rules allows to keep your data consistent
- Can serve as environment for other platforms
- Works well with microservices architecture

Technically - how to save events

NOSQL Data Storage	Special Event Store	JSON Friendly Relational DB	Backendless Stores
Raven DB	EventStore	SQL Server 2016	FireBase
Mongo DB	NEventStore	PostgreSQL	Parse
Cosmos DB		...	...
...			

CQRS + Event Sourcing

Event Sourcing together with CQRS



Real Life samples of CQRS applicability

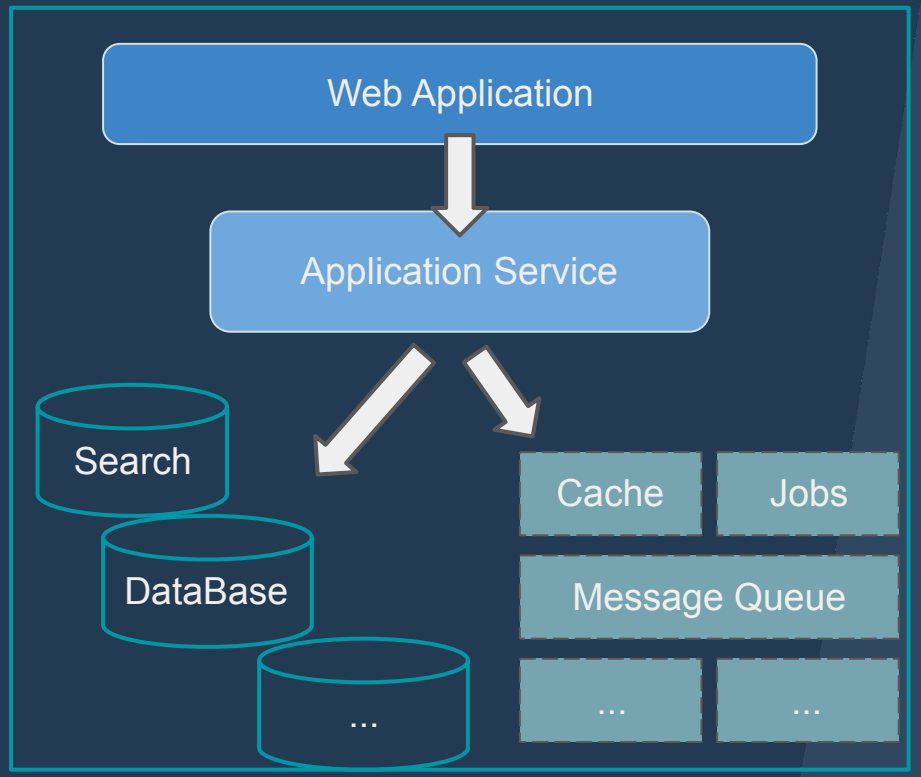
And event sourcing as well

Projects based on these approaches

1. Sample of Social Networks
2. Sample of Trucks Telematic project with event sourcing
3. Sample of Reporting for high load once a year for CQRS
4. Sample of Electric Station for event sourcing
5. Sample of money transfer for CQRS

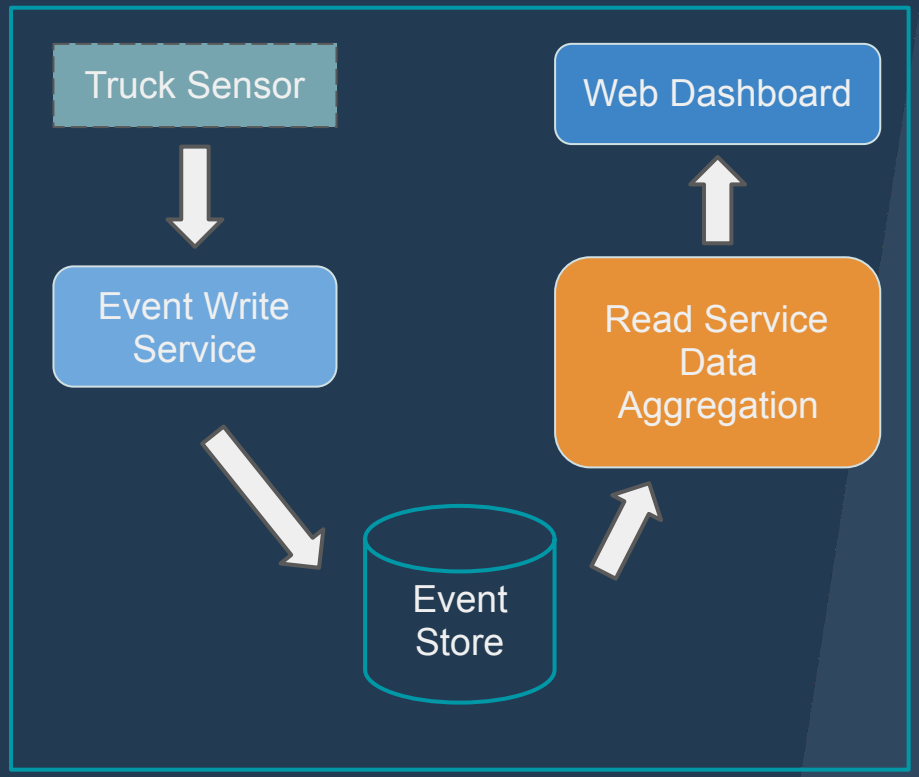
Social Network Sample

- When user posts something - app produces number of work items
  - Add data to event storage
  - Add view storage for user
  - Notify users (in browsers, push notifications….)
  - Update search indexes...



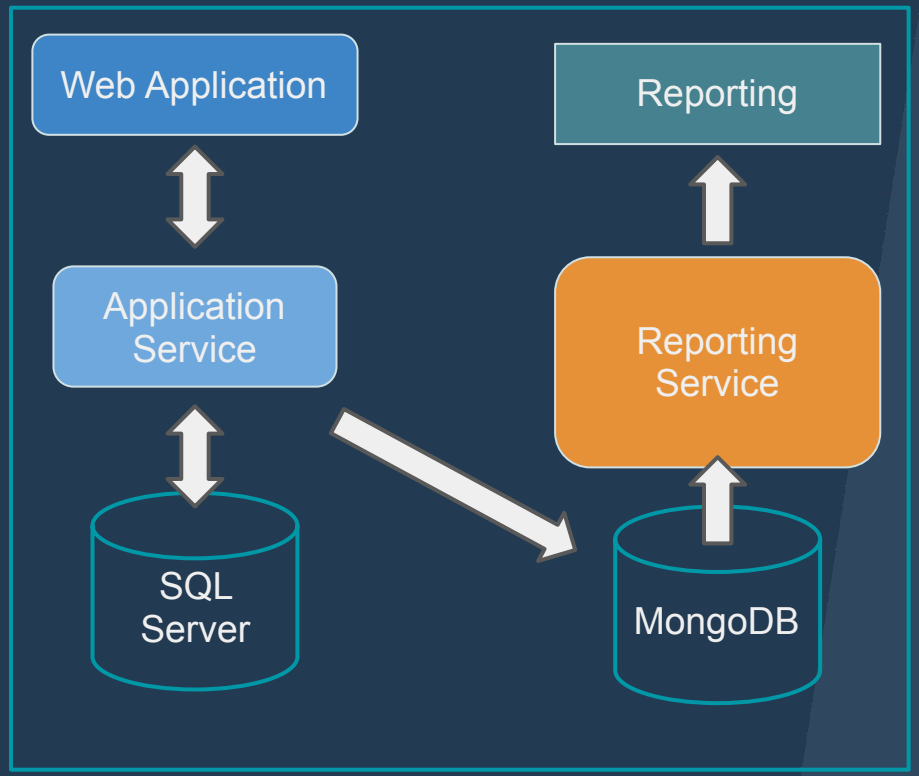
Trucks Telematics

- Eco System based on real time data from trucks
  - Speed, location, temperature, …
  - Engine turned on / off, …
  - Diagnostic messages
- Dashboard with average statistics and ability to change level of details
- Reporting



Finance Year Reporting

- System to enter finance data and see reports for government
- Data is entered all the time, but at the end of finance year application load increases 10 times
- Reporting should be available immediately



Electric Station

- Real app data - events from turbines, solar panels.

  Each impulse of data - event.
- No user interface to enter data
- No presentation layer to observe data
- Workers need to see some data model

Q&A

Thanks! Let’s discuss questions

Speaker: Valentin Kononov

Email: valentin.kononov@gmail.com

Skype: valentin_kononov

Links

- https://dev.by/lenta/oxagile/intervyu-s-avtorom-programming-microsoft-asp-net-mvc-dino-esposito
- http://neventstore.org/
- https://habr.com/post/146429/
- https://habr.com/post/149464/