ROAD TRAFFIC ANALYSIS
AND AGILE ARCHITECTURES
<table>
<thead>
<tr>
<th>Climate</th>
<th>Chemicals</th>
<th>Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snow plowing</td>
<td>Road salting</td>
<td>Vehicle count</td>
</tr>
<tr>
<td>Studded tires</td>
<td>Sea spray</td>
<td>Vehicle weight</td>
</tr>
<tr>
<td>Frost heave</td>
<td>Bird droppings</td>
<td></td>
</tr>
</tbody>
</table>
Responsible for state and county roads
Planning, construction, operation
7500 employees
Yearly spending: 3.6 Billion USD
**SYSTEM PURPOSE**

- **Reports**
  - Maintenance plans
  - Measuring changes
  - Weather impact

- **Real-time**
  - Expected travel time
  - Shape traffic flow
  - Emergencies

- **Forecasts**
  - Traffic forecast
  - Advise on routing
  - Shape commuting habits
Roadside equipment
Speed: loop distance / signal offset
Length: speed \cdot signal duration
Type: signal shape analysis
event_number: 2319762
utc_timestamp: 2016-03-16T23:58:44.330Z
lane: 1
speed: 80.9
length: 16.46
vehicle_type: BUS
3G Router
Data logger
Systems architecture
Building the system
FIRST TAKE: CONSTRAINTS & DESIGN IMPACT

Infrastructure:
- Over-the-fence deploy
- No access to production machines
- New stack for OPS team
- Few servers available for us
  - Few deployment artifacts
  - Include metrics in the application
  - Minimal complexity
  - Run on few nodes

Software design principles:
- Ensure control and transparency
- Utilize existing knowledge
- Fitness to purpose
- Maximize scalability
  - Libraries, not frameworks. No DI
  - Libraries we had experience with
  - Understand the high-level concepts
  - Study storage options
Needs:
- Receiving, processing and storing events
- Asynchronous calls
- Periodic calls
- Robustness, retries
- Redundancy

Building blocks:
- Message passing?
- Queue system?
- Scheduler?
- Actors?
- Cluster?

Akka Cluster
First Take: Storage

Needs:
- High event rate
- Store all events
- Calculate statistics
- Provide on-demand reports
- Robustness, redundancy
- No updates

Desired properties:
- Write scalability
- Volume scalability
- Counts, grouping, averages, percentiles
- Very low latency
- Distribution
- Append-only

Elasticsearch
Data loggers

Real time traffic view

Node 1

Node 2

Node 3
Push from A to B or pull from B to A?
Synchronous or asynchronous?
Peer-to-peer or via a broker?
What if A crashes? Or B?
Must data be validated?
Which data format?
Which protocol?
Authentication?
Performance?
Encryption?
Scalability?
Logging?
...
Desirable capabilities:
- Support for many types of devices
- Polling, triggers
- Real time event push, historical events
- Method calls
- Robustness
- Bandwidth efficiency
- Authentication
- Available vendors & implementations

OPC UA
Running the system
SECOND TAKE: REQUIREMENTS & IMPACT

Infrastructure:
- Scale out
- Dedicated clusters
- Configuration management
- Robustness

Application:
- Scalability – number of data loggers
- Reduced failover time
- More servers
- Installation control – ie scripts
- Deployment pipeline
- Using multiple data centers
- Distributed state
- Connection sharding
Gaining experience
Append only? Not quite.
- Compensate for bugs
- Compensate for uncovered cases
- Reformat old data to new spec

Reindexing data in ES is costly

Considering moving to classical “primary storage + ES”
- Need more nodes (another cluster) and more software
- Need more storage space
- Increases complexity
Near disaster
Select a time interval – for example, a month.

Divide the vehicle events within this interval into:

- 5 length categories, and provide counts and avg speeds for each category,
- 12 speed categories, and provide counts for each category,

....and also provide:

- the vehicle count and average speed for the entire interval,
- the 85 and 95 percentile speeds within the interval,

...and do this for:

- each lane being measured by a data logger
- and each data logger within a given region

On demand.
More and more data loggers
Increasing transaction rates
Creating the report takes a lot of RAM

Index corruption: Random data loss in some shards
Scope: 20% of all data from three previous months
YES
POST MORTEM

We recovered, somewhat incidentally

OOM should not lead to index corruption
Distributed consensus is hard to get right
See “Call be maybe” by Kyle Kingsbury
Elasticsearch has improved

No product is 100% reliable
Your design must compensate
Backup, cluster redundancy, upstream storage
Other hindsights
SOME THINGS CHANGE, SOME MUST STAY

Agile means optimizing for change

One thing stays: The need for correctness

Need to know why the code looks like it does

- Don’t document code, document the domain logic
- Exception: Non-intuitive match between code and domain

Need to know why the system state has become like this

- Use versioning / change logs: When, what, who

Take little signs seriously

- Why did that happen?
DATA LOGGERS ARE PHYSICAL DEVICES

Cannot be copied around
- Software implementation: Realistic?

Vendors interpret things
- Create a test kit for verification

Be agile when buying!
- Buy small batches, learn, adjust

Make them upgradable!
- Bidirectional integration

Have life cycles
- Repairs, maintenance, failures, replacement
REAL WORLD PHENOMENA
Agile architectures – our perspective
THE NEED TO BE AGILE

Develop → Deploy → Experience

Optimize
CHANGE GEARS ALL THE TIME

risk

system property A

system property B

system property C

time
feature change

technical change

DO NOT BUNDLE
Aim for few dependencies, ie isolation
Aim for feature richness, within a narrow domain
Goal: Grow within what you have
Accept you might break the Minimal Viable Product principle
Aim for expressiveness in protocols or formats
Know your computer science theory!
SUMMARY

Pay attention to the constraints
Think long-term, act immediately
Create a compatibility kit
Ensure devices can be remotely upgraded
QUESTIONS? COMMENTS?

Kristoffer Dyrkorn, BEKK
@kristofd
kristoffer.dyrkorn@bekk.no
Screenshots
# 200820 - LYSAKERLOKKET

## Trafikk på målestasjon

<table>
<thead>
<tr>
<th>Trafikkstasjon</th>
<th>Hastighet (km/t)</th>
<th>Avstand (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YHF FRA OSLO</td>
<td>56</td>
<td>3.5</td>
</tr>
<tr>
<td>MF FRA OSLO</td>
<td>79</td>
<td>3.9</td>
</tr>
<tr>
<td>VF FRA OSLO</td>
<td>70</td>
<td>4.3</td>
</tr>
<tr>
<td>VF FRA DRAMMEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MF FRA DRAMMEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HF FRA DRAMMEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>YHF FRA DRAMMEN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Motretning: Drammen