Engineering Automotive Systems
a visionary outlook

AUDIOWIDE - REGULAR 36
Introduction
Goals and non goals
Engineering northstar and areas
Engineering systems
Engineering design areas
Goals and non-goals of this talk

- **Goal is**
  - To showcase software engineering practices and designs to match the vision of Eclipse SDV
  - Understand what can be leveraged for automotive cloud and vehicle solutions
  - Understand the benefits of a cloud native approach and provide food for thought to dive deeper into topics

- **Non goal is**
  - Discuss solution architectures
  - Discuss all aspects of engineering processes and principles (we only have 30 minutes)
  - Provide a final solution/answer to everything (we already know it’s 42)
Automotive guy and software guy

“We need to become a software company

“Hey we do have 300 services that deal with everything from compute over networking all the way to data. We have so many services that are similar in functionality, so there is certainly something there for automotive”

“Great, we are planning to connect about 100 million devices and need to serve multiple tenants while honoring regulations and by the way we need to transform onboard architectures to mimic cell phones”

“Does everything in automotive need to be complicated!”
The engineering northstar

leverage same process
leverage same technology
leverage same tools……
…whenever possible

while ensuring safety
# Two different worlds

<table>
<thead>
<tr>
<th>Category</th>
<th>Traditional</th>
<th>Cloud Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality of code check-ins</td>
<td>Depends on scenario</td>
<td>Validated through unit tests</td>
</tr>
<tr>
<td>Environment Creation/Configuration</td>
<td>Manual</td>
<td>Automated</td>
</tr>
<tr>
<td>Deployment Frequency</td>
<td>1-2x a month (or less frequent)</td>
<td>Deploy whenever needed, including several per day</td>
</tr>
<tr>
<td>App Deployment Process</td>
<td>Requires meetings and planning</td>
<td>Push-button deployment</td>
</tr>
<tr>
<td>Deployment validation</td>
<td>Manual</td>
<td>Automated</td>
</tr>
<tr>
<td>Observability</td>
<td>Minimal to none, mainly monitoring</td>
<td>Observability of the entire stack</td>
</tr>
<tr>
<td>Dev and Ops relationship</td>
<td>Blame culture</td>
<td>Culture of trust</td>
</tr>
</tbody>
</table>
Engineering principles that should be considered (excerpt)

● **Organizational challenges**
  ○ Small team with bounded context
  ○ You build it, you run it

● **Automation**
  ○ IaC
  ○ Gitops
  ○ Gated Rollouts

● **Observability**
  ○ Monitor the entire stack
  ○ Correlate events
  ○ Use the same standard, e.g. OpenTelemetry

● **Testing**
  ○ Shift to a testing pyramid (shift left)
  ○ Testing services in isolation does not offer much value
  ○ Decide between safe and fast

● **Service Design**
  ○ Use tools like OpenAPI
  ○ Consider serialization costs
Deployment velocity: Safe vs Fast

- **SAFE**
  - Follows traditional automotive development
  - ASPICE/ISO
  - Less frequent component update
  - Safety/security critical components
  - Artifacts binaries
  - In car only

- **FAST**
  - Follows cloud native processes (DevOps, GitOps)
  - Queued and staged deployments
  - Frequent component update
  - Non safety/security relevant components
  - Cloud and car
High level engineering system to support vehicle and cloud

Idempotent Operations

Artifact Repo

slow

comfort

deploy

*aC

fast

*aC

Transform

Frontend

deploy

Idempotent Operations

COPYRIGHT (C) 2023, ECLIPSE FOUNDATION. THIS WORK IS LICENSED UNDER A CREATIVE COMMONS ATTRIBUTION 4.0 INTERNATIONAL LICENSE (CC BY 4.0)
On board system zoom in
Important engineering design principles that can be translated to automotive

Moving into “SOA” means moving into distributed systems

<table>
<thead>
<tr>
<th>Engineering design principle</th>
<th>Cloud</th>
<th>Vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loose Coupling</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Fault tolerance</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Scalability</td>
<td>Y</td>
<td>N*</td>
</tr>
<tr>
<td>Consistency</td>
<td>Y</td>
<td>In car**</td>
</tr>
<tr>
<td>Security</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Performance</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Interoperability</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

*Currently only process/node
**Strong consistency not required
THANK YOU!
Engineering building blocks – cloud native perspective

- **Project/Program Management**
  - Agile/Scrum
  - Work Item Tracking

- **Development**
  - Developer Experience (DevEx)
  - Design Reviews
  - Code Reviews

- **Testing**
  - Executed during CI/CD as pyramid

- **Operations**
  - Observability
  - Source Control

![Diagram](image-url)