From Zero to Useless to Hero

Make Runtime Data Useful in Teams
Contact us if you want. =)

Robert Hoffmann
Lead Architect VPaaS
{name.surname}@telekom.de

Dr. Florian Lautenschlager
Software Architect
{name.surname}@qaware.de
“Hallo Magenta”

Building a European Voice Assistant Platform
From Zero to 1 international co-development

> 900 collaborators

> 500 active git repos

> 100 services
Advanced toolchain needed. Standard used.

Metric sampling-based
- Exploration
- Probes, Collection and Storage
  - Grafana
  - Prometheus

Textual event-based
- Storage, Exploration
- Transport
  - elastic
  - fluentd

Span event-based
- Storage, Exploration
- Probes and Collection
  - ZIPKIN
  - Spring Cloud Sleuth

NEW:
- Grafana Cloud
- Humio
- Jaeger
- Cloud
STANDARDIZE ALL THE THINGS
Generic-Standard-Runtime-Data-Smarthub-Service-Data-Model

- **Logging Concept**
  - Log-Message
    - Timestamp: Timestamp
    - Service: String
    - TraceId: String
    - Response: String
    - ...
    - + log(Context): void

- **TraceId in every Response**

- **Standard Metrics for incoming and outgoing Requests**
  - Incoming-RED-Metric
    - name: String
    - value: numeric
    - tags: [tenant, service, call, target, code, ...]

- **Outgoing-RED-Metric**
  - name: String
  - value: numeric
  - tags: [tenant, service, call, target, code, ...]

- **Standard Readiness and Liveness checks with Metrics**
  - Readiness/Liveness-Check
    - + erroneousChecks: Integer

- **Tracing Best-Practices**
  - Trace
    - tenant: String
    - userId: String
    - testing: Boolean
    - annotations: [DB-Calls, etc..]
    - ...
    - + trace(Context): void

- **Standard Database Metrics and specific ones.**
  - Database-Call / ...
    - name: String
    - value: numeric
    - tags: [tenant, repository]
“Done. This solves all our problems. They will love it!“

– We, the ignorant ones
Our team: Colorful.
Our solution: Monochrome.
T=Useless

Because we are monochrome
Nobody wants to be a Beginner. Optimize for Intermediate.
What we did to move our solution from expert to intermediate.

Useful = Utility + Usability. 😐

Utility: whether it provides the features you need.

✅ You can find all the information...

Usability: how easy & pleasant these features are to use: Learnability, Efficiency, Memorability, Error Handling, Satisfaction.

❌ ... if you really know how and where to look (as an Expert).

Usability 101 - Jakob Nielsen

https://www.nngroup.com/articles/usability-101-introduction-to-usability/
Close Gaps: Link data and tools as much as possible.

Dashboards with links to **logs** and **e2e test runs**
Close Gaps: Link data and tools as much as possible.

<table>
<thead>
<tr>
<th>admin_management</th>
<th>conversation_history</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Status</strong></td>
<td><strong>Status</strong></td>
</tr>
<tr>
<td>19-12-11 16:14</td>
<td>19-12-11 16:15</td>
</tr>
</tbody>
</table>

**Pipeline UI - promote software and get runtime data**
Close Gaps: Link data and tools as much as possible.

Pipeline dashboards with logs, traces
Close Gaps: Link data and tools as much as possible.

Welcome Robert Hoffmann

Useful links
- Kubernetes Dashboard
- Grafana
- Jaeger
- Kibana
- Kibana Conversation History
- Prometheus

Search by Traceld in Jaeger/Kibana
Insert Traceld and either click Jaeger or Kibana.
Traceld:
5f4c09efbe83171

Search Jaeger
Search Kibana

Search by Userld in Jaeger/Kibana (only for CVI-Core)
Insert Userld and either click Jaeger or Kibana.
Userld:

Gangway landing page to access **k8s, logs, traces, metrics**
Make functional use: First-level support integration.

**GDPR-aware debugging in production:**
Token-based user-specific debug logging and tracing
Make functional use: Resolving Tickets more easily.

Referencing Trace IDs as a common base to discuss and find relevant data
Lower the access hurdle: CLI & Chatbot integration.

Any project member has **easy access**. Just open your chat.

Anyone can **learn by example**. See how others use the service.

Support in case of an **error**. By others or technical:

- **Trace**: Request Trace
- **Logs**: Request Application log
Changes in the culture that we have recognized.

**Visibility and Increased Trust**: Toolchain acts as a safety-net as it shows the runtime behavior. *People can be sure to understand their services, e.g. in case of an error.*

**Self-Awareness**: Accept and understand that software has a runtime behavior. *Not all developers feel comfortable with dynamic analysis, but now they have means to see and understand.*

**Clear Communication**: Inner & cross-team communication is easier. *Different people can easily share the same context, e.g. trace-Id, log messages, request flow.*

**Error Culture**: Failures are more easily accepted. *As the software system is visible and the cross-team communication is clear, people tend to accept failures and work together on solutions.*

**Ownership**: Increased acceptance is the foundation for end-to-end responsibility. *Due the disability and increased trust, clear communication and error culture, people are more inclined to take ownership for their services.*
Start here
Select Toolchain & Standardize Metrics, Logs, Traces
Link and combine them as far as possible
Integrate them into everyday tools & Processes
Tools your team