# Building Cloud Native GDPR Friendly Systems for Data Collection

# What is VTT?

- Technical Research Centre of Finland
- About 2000 researchers
- Wide array of topics
  - Nuclear safety
  - Printed electronics
  - Food science
  - Data-driven services
- About 200 projects develop software yearly, involving 5% of personnel
- Yearly 10-20 projects have to gather new datasets for research

# GDPR in a nutshell

# GDPR in a nutshell

- General Data Protection Regulation
- Came into effect on 25th of May, 2018
- Contains rules for protection, privacy and processing of personally identifiable data of EU / EEA citizens regardless of the place of processing
- Defines the rights of individuals
  - Transparency about the data handling process, data breaches, etc.
  - Access to personal data
  - Correct / delete personal data
  - Etc.

# GDPR in a nutshell (cont.)

- Adapting these rules required changes on many levels of the organization
  - Improved data management and access control
  - Company DPO
  - GDPR Handbook for project managers
- Projects play a very important role too
  - Data-mapping
  - Impact analysis

# GDPR in a nutshell – Data-mapping

- What data will be collected (hardest question for research)
- Check if any personally identifiable data will be collected
- Define the basis for data collection:
  - Consent
  - Contract
  - Public interest
- Define the data security features:
  - Transport / storage / archival security
  - Pseudonymization or anonymization
  - Access control

# How to help our projects?

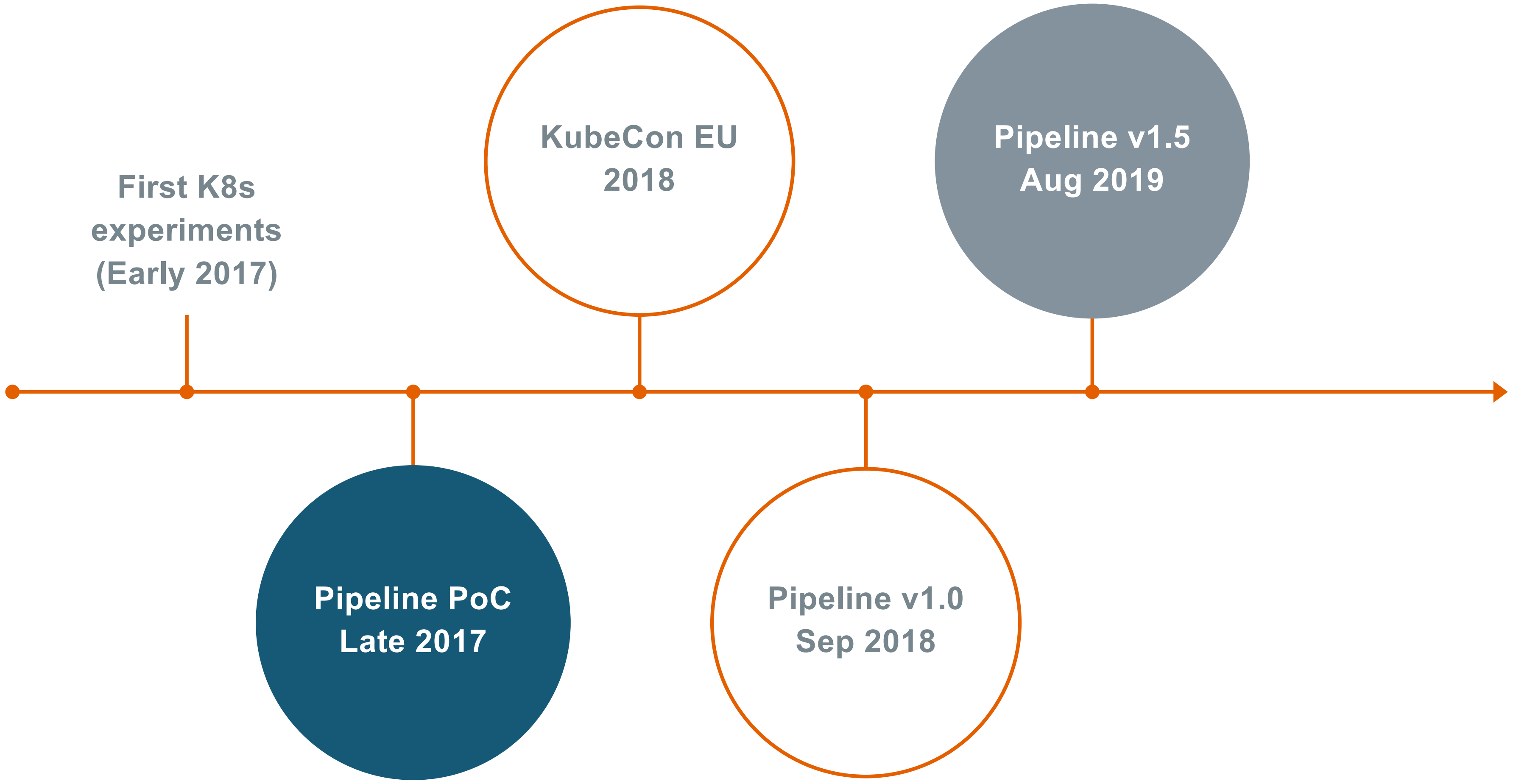
# A generic pipeline

- Project team has to own the deployment
- Empowering the researchers
  - They are experts of their fields (e.g.: machine learning)
  - The best way to use their talent is to do research
- We want to give them tools that takes care of the basics
  - Automated provisioning
  - Monitoring
  - Ingress with TLS*
  - Cluster-internal mTLS (between services)*
- Customizable

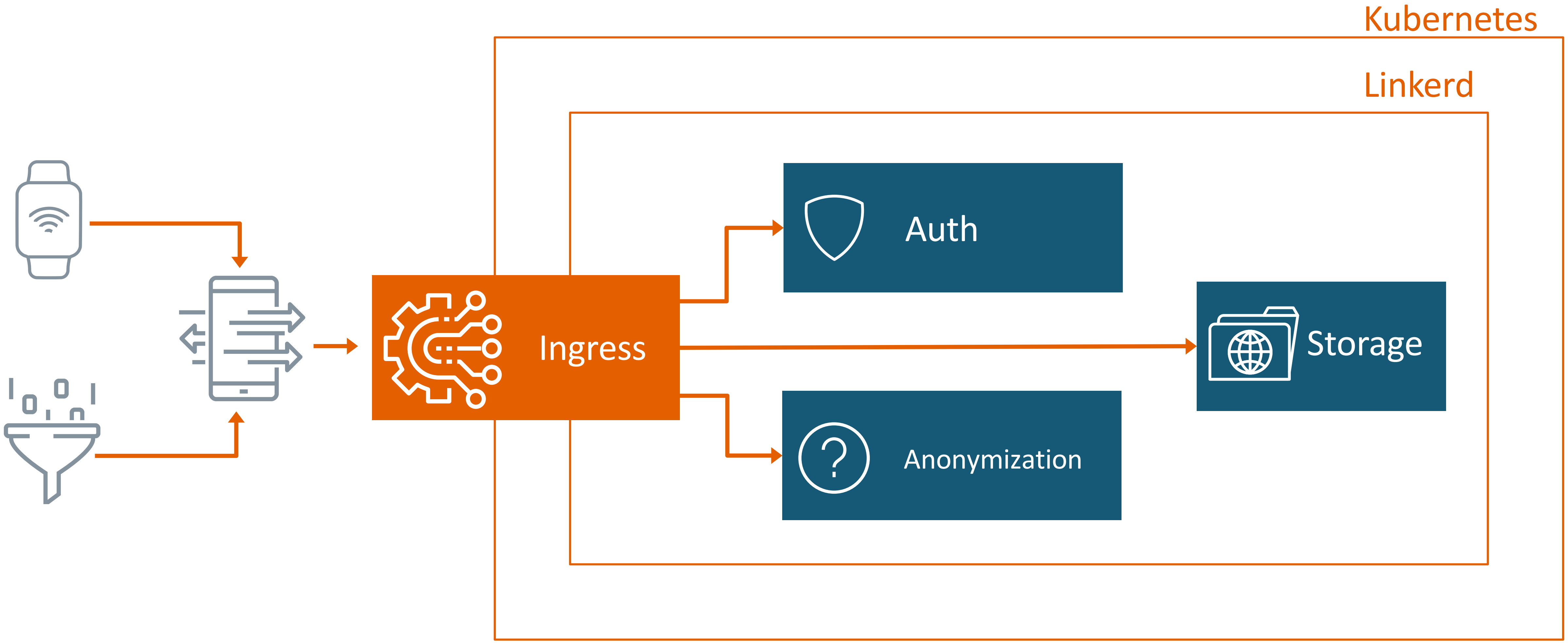
# A generic pipeline (cont.)

- We also provide some generic components (microservices)
  - Timestamped key-value store with optional location data, encryption*
  - Authentication / authorization service (uses OIDC, user ID tokenization)*
  - Location anonymization using machine-learning (trained on user-data to identify often visited areas)*
  - Pre-processing tools*
  - Android application to collect sensor data (“BT gateway”)
- Not a standalone project
  - Identify reusable components in public-research projects
  - Refine / extend iteratively

# Our timeline



# Pieces to the (cluster) puzzle


Kubernetes
Linkerd
Auth
Ingress
Storage
Anonymization

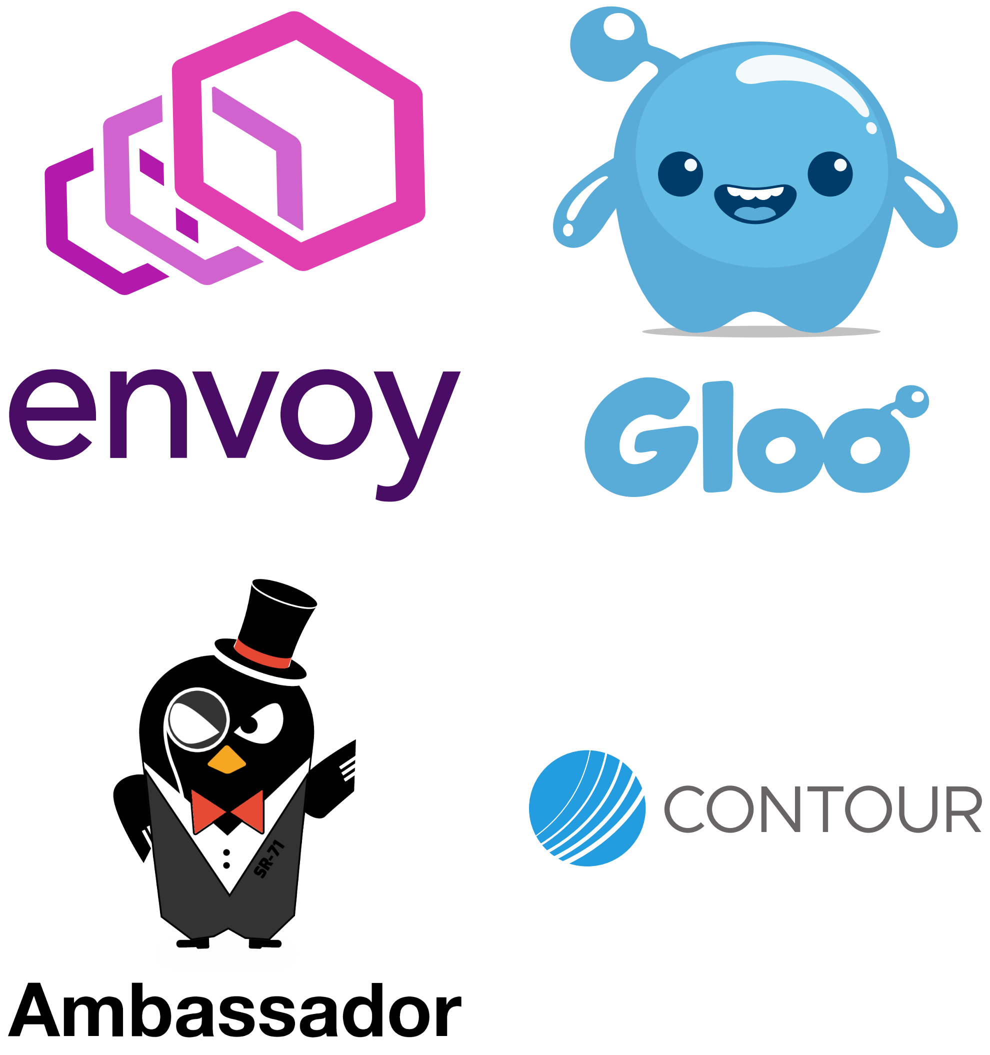
# RPC protocol

- Http/2 based RPC protocol
- Protobuf based data-object / service definition
- Client / server bindings are generated
- Many target languages
- Effective, binary data-representation
- gRPC-web brings support for web-clients

gRPC

# Ingress

- When using gRPC a LoadBalancer type Service is not ideal
  - Layer 4 vs Layer 7
- Takes care of TLS termination
- We had previous experience with Envoy, but other options are also available (e.g.: Nginx, Traefik)
- All of them offer features beyond Ingress specification



# TLS certificate management

- Certificates from Let’s Encrypt
- Cert-manager by Jetstack
  - Supports HTTP and DNS based validation
  - HTTP validation works only if Ingress objects work
  - Only DNS based validation supports wildcard domain names

# Service mesh

- Original goals:
  - Monitoring with no change to service code
  - Pre-configured dashboard
  - Lightweight (memory, CPU)
- mTLS originally seen as nice extra
  - With certain data types (sensitive personal information, e.g.: health data) it helps a lot with GDPR compliance
  - Some performance penalty
- Nice functions we don’t utilize much yet
  - Retry budget

LINKERD

# Automated provisioning

- Infrastructure as code, using real programming languages
  - JavaScript / TypeScript (Node.js)
  - Python
- Automatic and manual dependency
- Great Kubernetes support
  - Programmatic Kubernetes objects
  - Helm charts / Standalone Yaml files
  - Waiting for components to became ready

# Batch processing

- Argo Workflows
- Container-native workflow engine
- Multi-step workflows modelled as directed acyclic graph (DAG)
- Parallel steps
- Parameterizable
- Loops / conditionals
- Artifact support

# Demo time!

# Quick links

- TGIK: https://github.com/heptio/tgik
- K9S (Kubernetes CLI): https://k9ss.io/
- Kubernetes context switcher: https://kubectx.dev/

- Kubectl plugins
  - Package manager: https://krew.dev/
  - Access matrix: https://github.com/corneliusweig/rakkess
  - Wireshark: https://github.com/eldadru/ksniff

# Presenter info

- Zsolt Homorodi, Senior Specialist, VTT
- @HaZseTata
- https://github.com/hazsetata
- https://gitlab.com/hazsetata

- Demo: https://gitlab.com/hazsetata/kceu2019