



Eclipse eCAL - overview and deep dive "Distributed Recording"

What's that ? Eclipse eCAL™ ?



enhanced **C**ommunication **A**bstraction **L**ayer

Communication Stack

2016 – Continental Automotive R&D

2019 – OpenSource under Apache 2

2022 – Eclipse SDV Project

Prototyping AD vehicle systems



Programming Languages

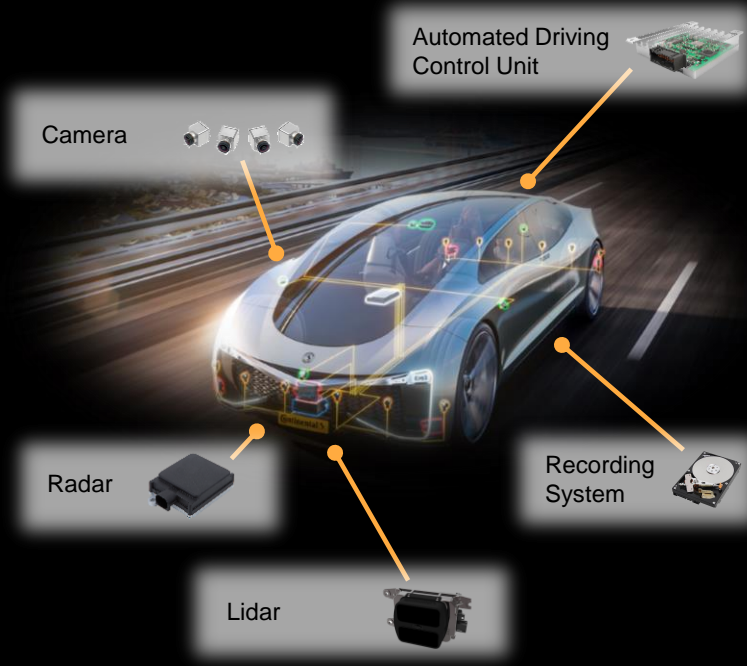
Operation Systems

Hardware Targets

Interfaces

Acceptance

Prototyping AD vehicle systems – Bonus level „Data Management“



Massive Data

High Speed Transmission

Distributed Computing System

Monitoring

Recording

How can eCAL handle this?

Designed for Multicore / Multihost Systems

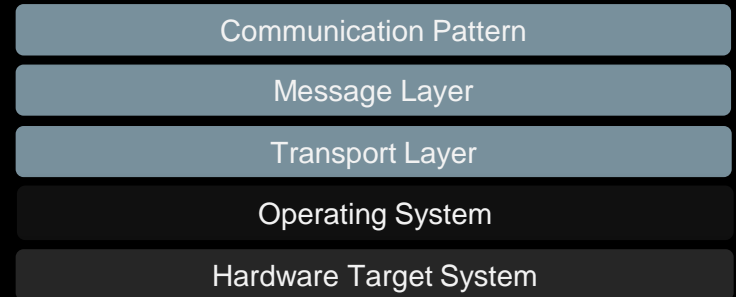
- **eCAL distributes everything**
 - Sensor Interfaces
 - Software Components
 - Ecosystem Applications
- **eCAL is highly optimized on performance**
 - C++ core
 - Interprocess - Shared Memory Support
- **eCAL has powerful tools**
 - Monitoring, Recording, Replay, System Boot/Shutdown
 - Distributed Recording Concept – unique selling point ☺



What's behind it? The architecture.

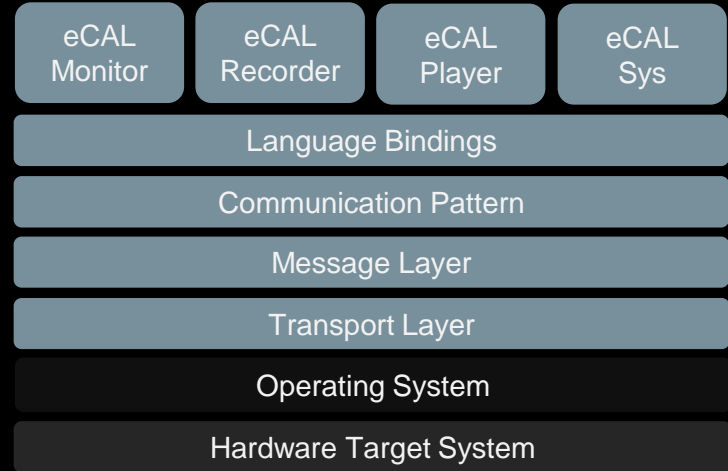
Architecture – Core

- supports **POSIX** as well as **Windows** operating systems
- supports **different transport protocols**
 - inter-process communication: **shared memory**
 - inter-host communication: **udp multicast / tcp**
- supports **different serialization formats**
 - google::protobuf
 - capnproto
 - google::flatbuffers, messagepack, json ..
- supports **publish / subscribe** and **client / server** pattern
- **brokerless dynamic design**



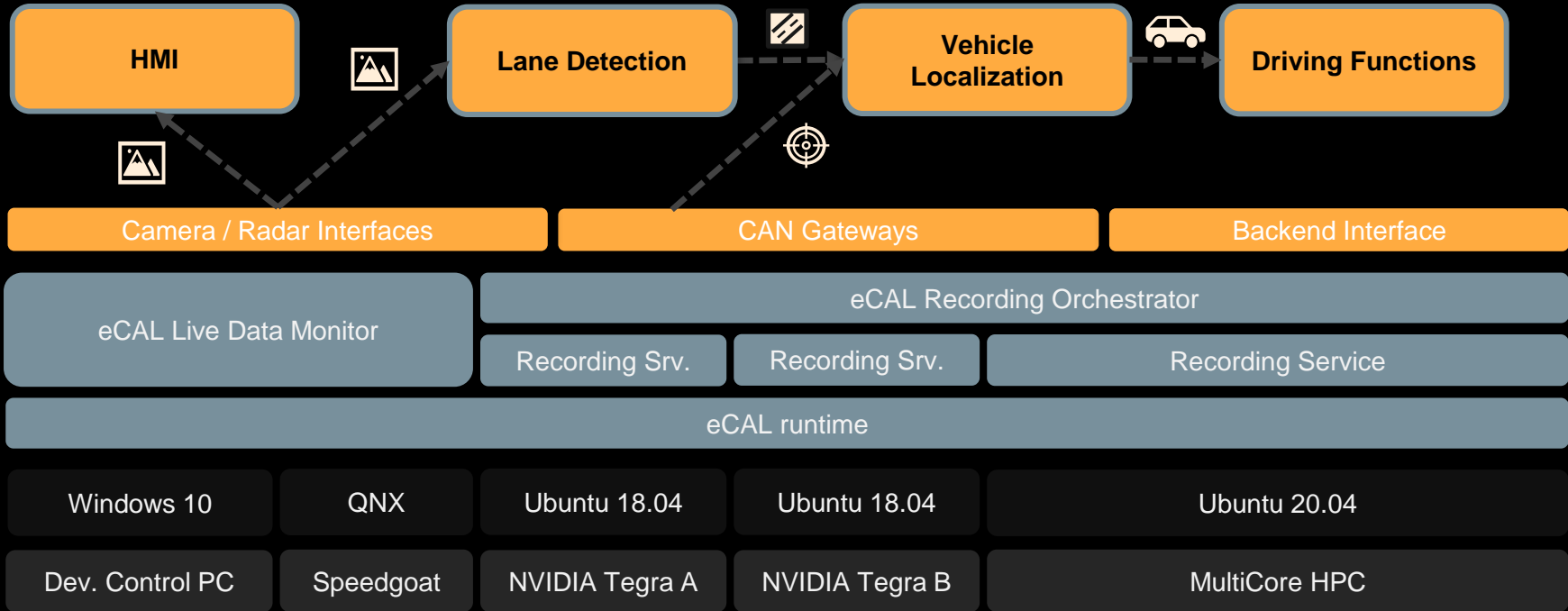
Architecture – Languages and Tools

- language bindings for C, C++, C#, Python, Rust, Go, M-Script, Simulink
- shipped with eco system tools for
 - live monitoring of all software component interfaces
 - orchestrated, distributed message recording
 - message replay real-time or stepwise
 - automated component start, stop and supervising
 - all tools realized as command line and GUI application

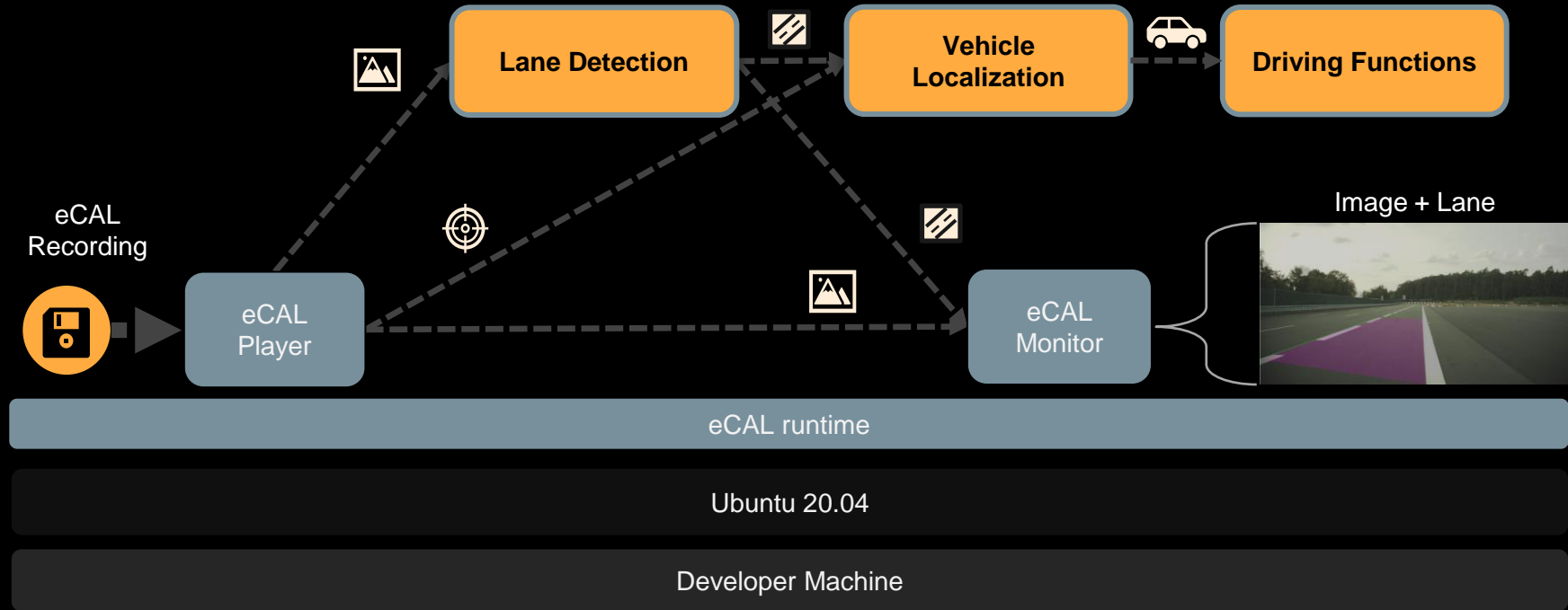


From theory to practice. Typical use cases.

AD Demo Vehicle Setup



Validation of 3 Software Components



Deep Dive “Data Recording”

AD Recording – Problem Statement



Throughput

Distribution

Postprocessing

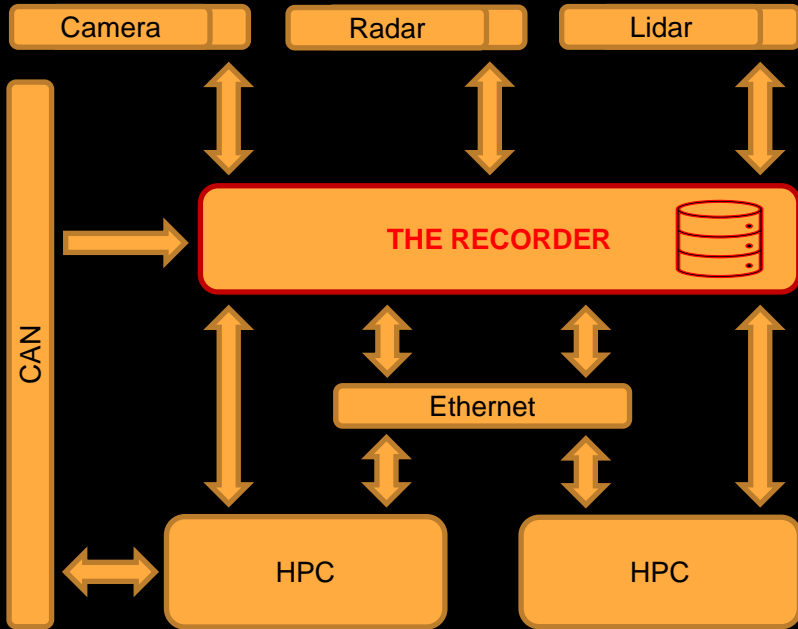
Persistency

API's



Classic Recording Approach

The magic, single (hardware) recording device



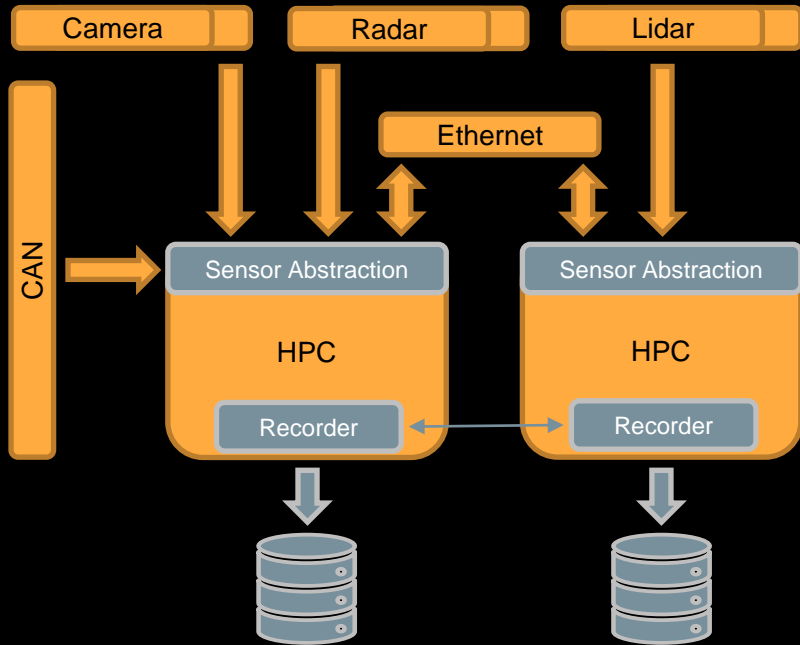
Disadvantages:

- Bottlenecks everywhere
- Expensive hardware
- Sensorics pass-through
- High configuration effort



eCAL's Distributed Recording Solution

eCAL's Distributed Recording Solution



Recording:

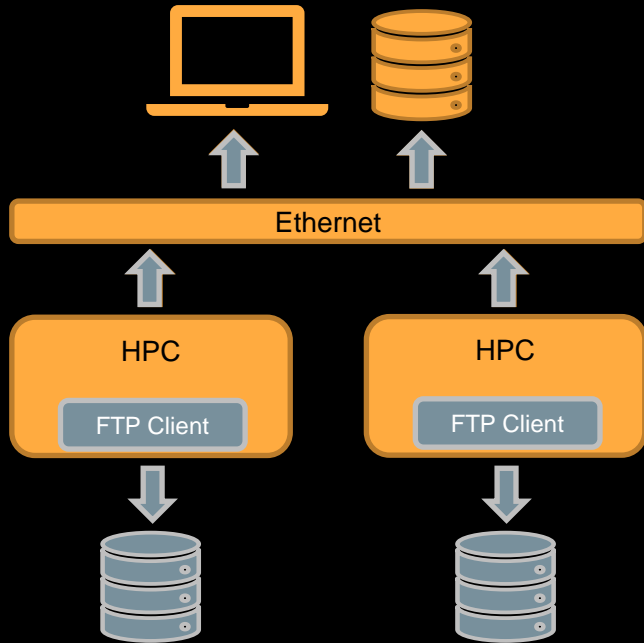
Interface abstraction

Distributed recording instances

Orchestration / synchronization

Scalability / cost reduction

eCAL's Distributed Recording Solution



Aggregation:



Integrated FTP clients

Automated reassembling

Different storage targets

Current format HDF5

Live Demo “Recording in Action”

Remember your questions 😊

Summary

Summary



eCAL is designed for autonomous driving applications

eCAL combines modern communication patterns with state-of-the-art message protocols

eCAL has powerful tools for rapid prototyping

eCAL is open source since 2019 and part of the Eclipse family since 2022 😊

Thank you!



Join the conversation:

 [@EclipseCon](https://twitter.com/EclipseCon) | [#EclipseCon](https://twitter.com/EclipseCon)

