The Design and Application of Baetyl
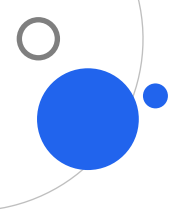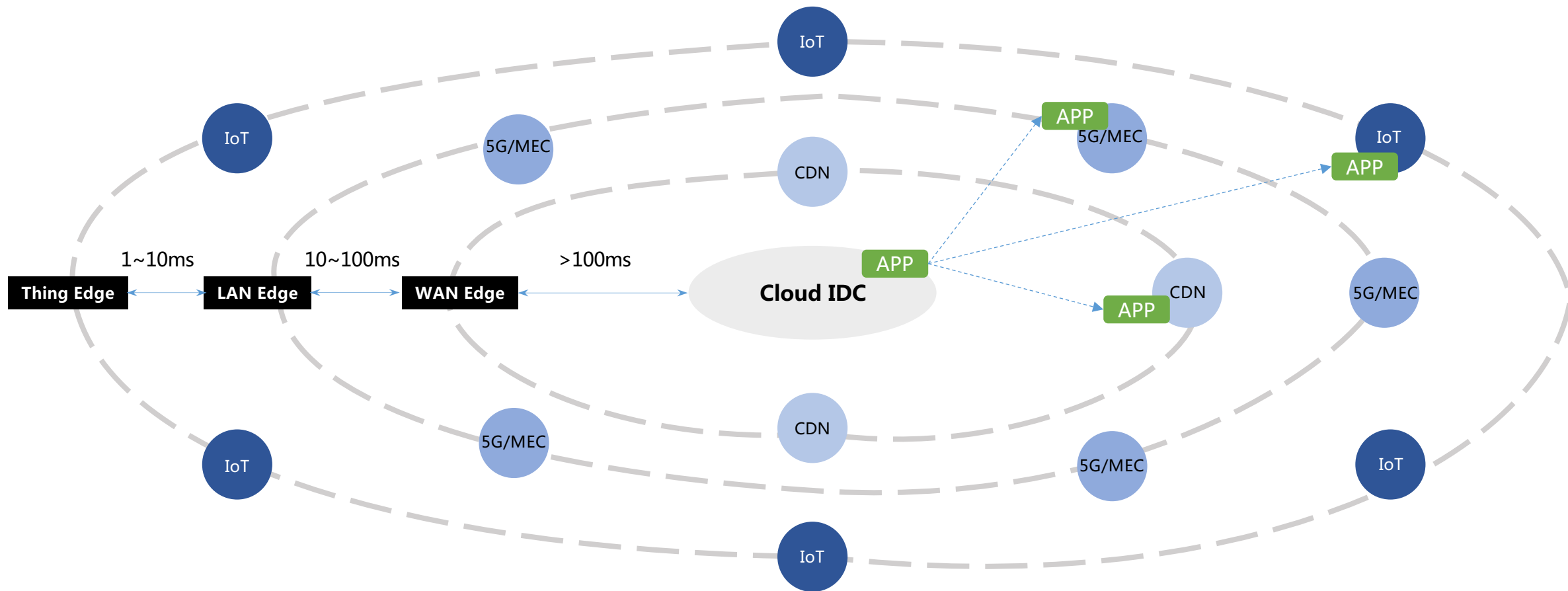
LI Leding
Chair of Baetyl TAC
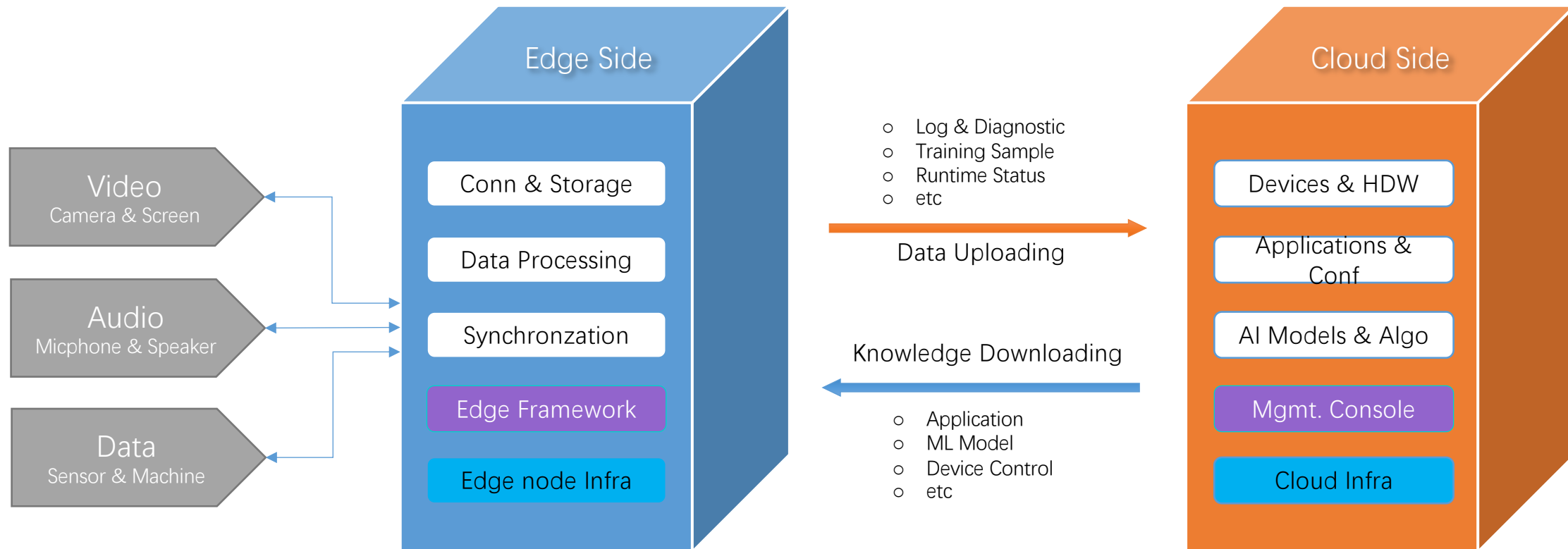
Jun 2022
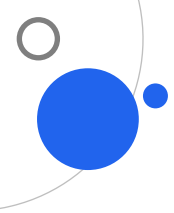
Computing is everywhere

Fusion between the Cloud and the Edge

Edge Side

- Video
  Camera & Screen
- Audio
  Micphone & Speaker
- Data
  Sensor & Machine

Conn & Storage

Data Processing

Synchronzation

Edge Framework

Edge node Infra

- ○ Log & Diagnostic
- ○ Training Sample
- ○ Runtime Status
- ○ etc

Data Uploading

Knowledge Downloading

- ○ Application
- ○ ML Model
- ○ Device Control
- ○ etc

Cloud Side

Devices & HDW

Applications & Conf

AI Models & Algo

Mgmt. Console
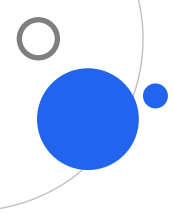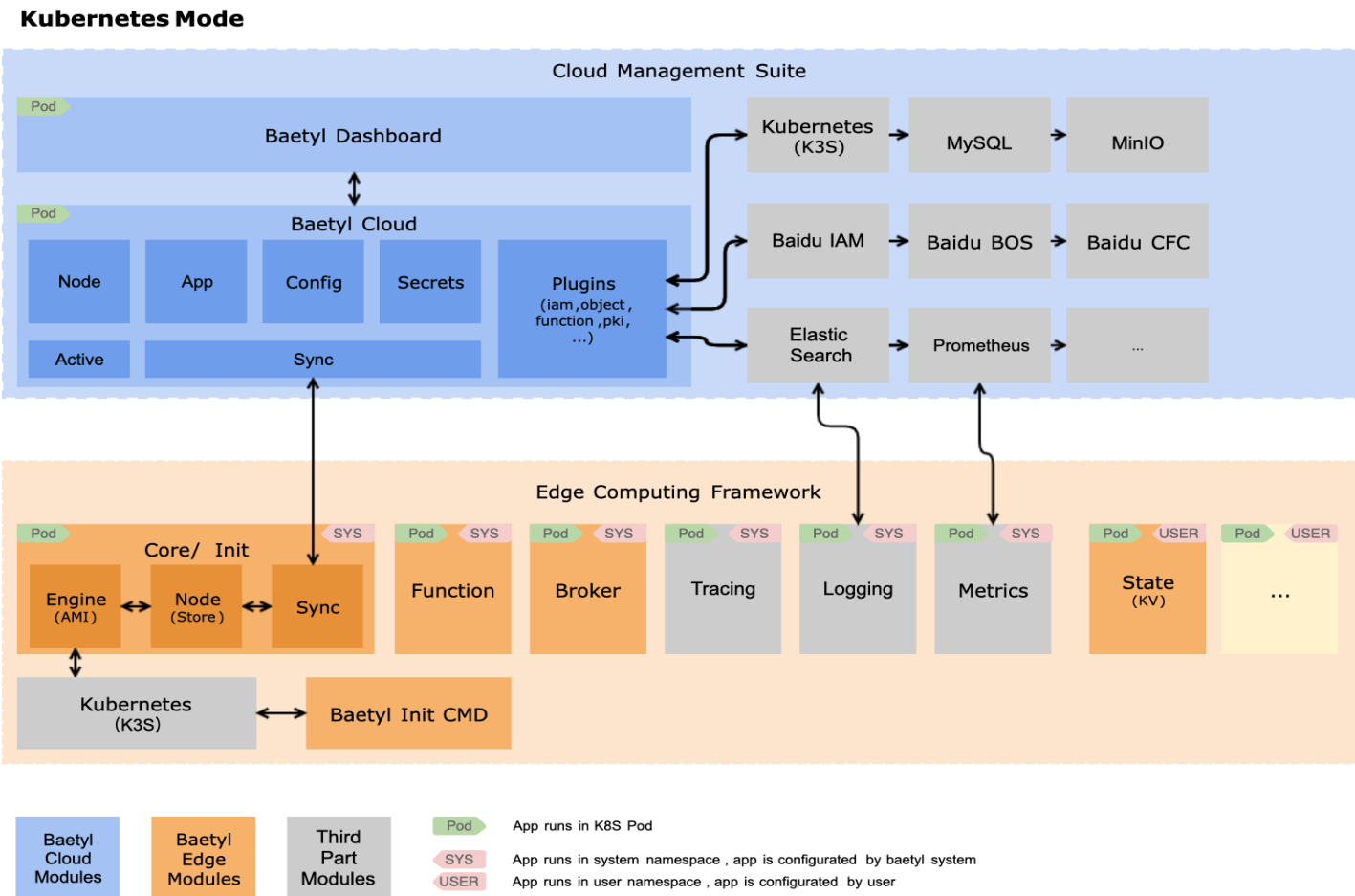
Cloud Infra

BAETYL

- Launched in 2017 as Baidu IntelliEdge, open source in 2018

- Join LFEdge umbrella in 2019

- Provide Cloud Native environment for all applications and AI services

- Unified management for both cloud and edge

- Unified security model and data privacy protection

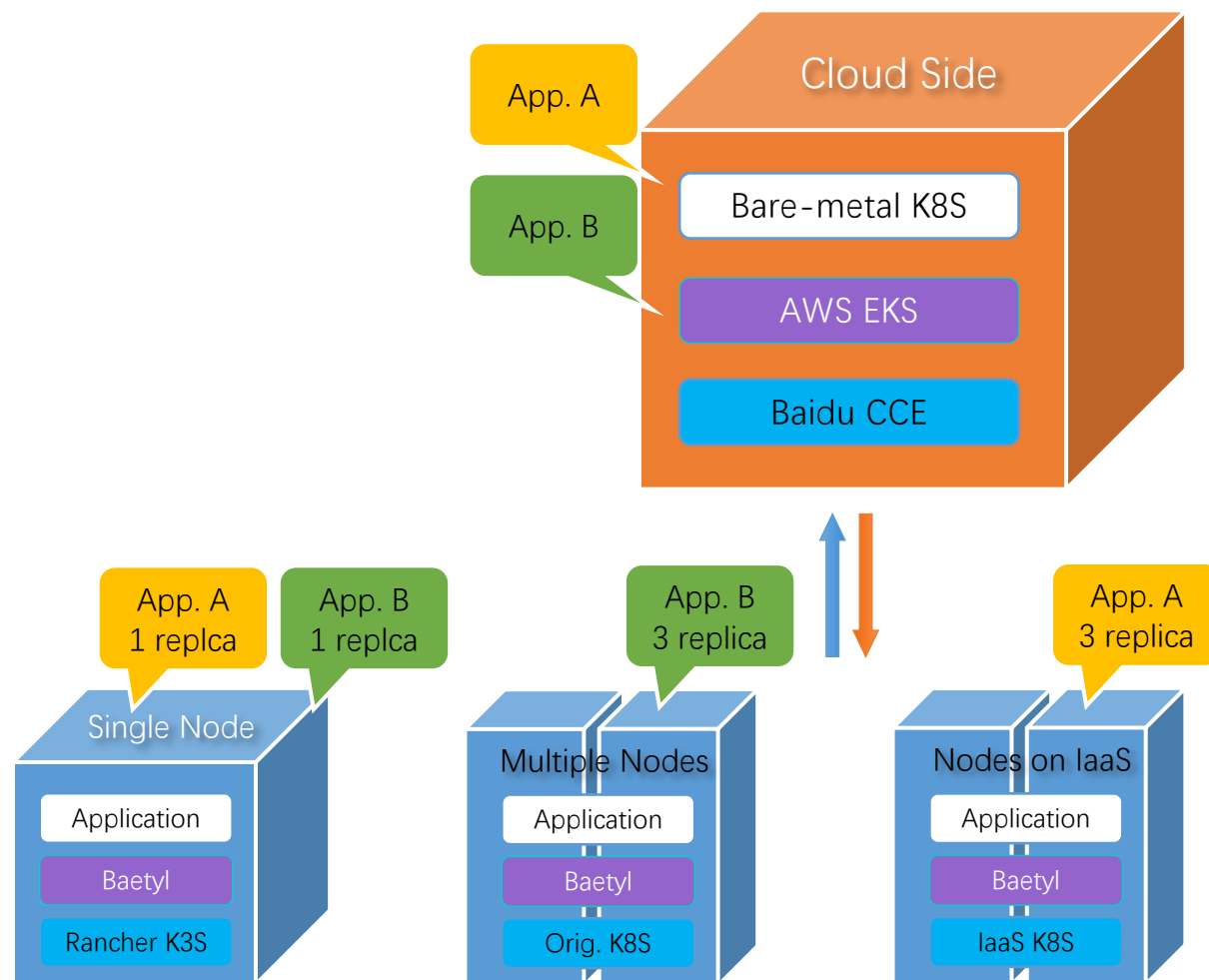- Vendor neutral, Apache License

https://baetyl.io/

Baetyl Architect

- Leverage Kubernetes

- Edge cluster, multiple node

- Open-source management
  - baetyl-cloud
  - with OpenAPI

- Self upgradable
  - Baetyl is running as a Pod

- More integrated services
  - Local KV storage
  - Logging & Tracing
  - Remote metrics report

Kubernetes Mode

Cloud Management Suite

Pod — Baetyl Dashboard

Kubernetes (K3S) → MySQL → MinIO

Pod — Baetyl Cloud

Node | App | Config | Secrets | Plugins (iam, object, function, pki, ...)

Active | Sync

Baidu IAM → Baidu BOS → Baidu CFC

Elastic Search → Prometheus → ...

Edge Computing Framework

Pod SYS — Core/ Init
Engine (AMI) ↔ Node (Store) ↔ Sync

Pod SYS — Function
Pod SYS — Broker
Pod SYS — Tracing
Pod SYS — Logging
Pod SYS — Metrics
Pod USER — State (KV)
Pod USER — ...

Kubernetes (K3S) ↔ Baetyl Init CMD

Baetyl Cloud Modules
Baetyl Edge Modules
Third Part Modules

Pod — App runs in K8S Pod
SYS — App runs in system namespace, app is configurated by baetyl system
USER — App runs in user namespace, app is configurated by user

Open ecosystem, Vendor neutral

☐ A Cluster-to-Cluster design
  - ✓ inspired by K8S Federation
  - ✓ self-scheduling
  - ✓ local load balance & failover

☐ No *modification* to kubelet
  - ✓ no specified K8S vendor requirement
  - ✓ no specified K8S version requirement

☐ Full K8S semantics in edge instance
  - ✓ Pod, Deployment & Service
  - ✓ DaemonSet
  - ✓ ConfigMap & Secret

Cloud Side

App. A
App. B

- Bare-metal K8S
- AWS EKS
- Baidu CCE

App. A 1 replca
App. B 1 replca
App. B 3 replica
App. A 3 replica

Single Node
- Application
- Baetyl
- Rancher K3S

Multiple Nodes
- Application
- Baetyl
- Orig. K8S

Nodes on IaaS
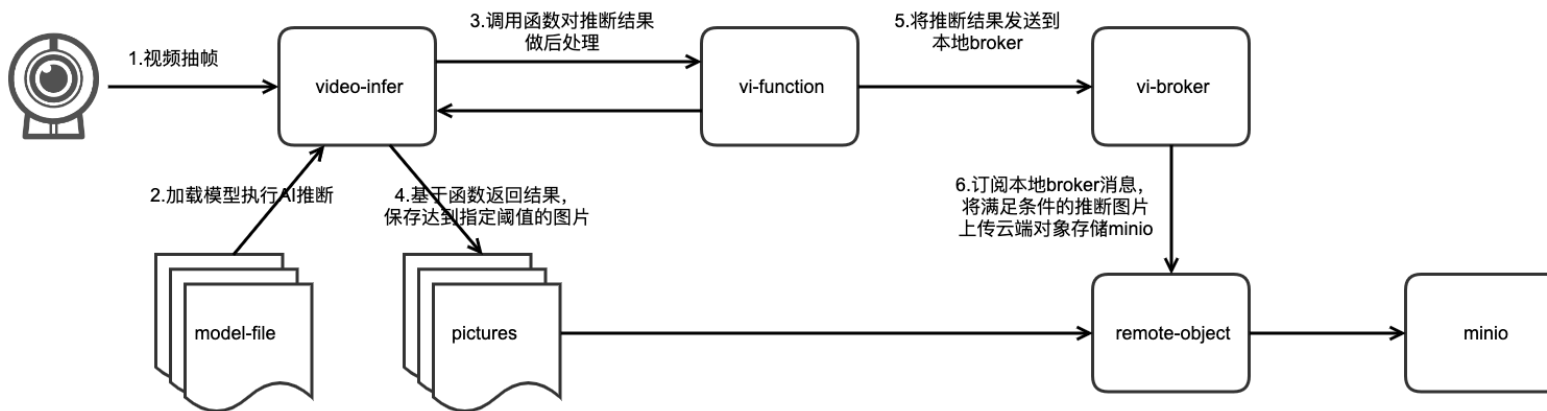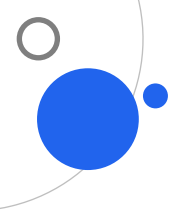- Application
- Baetyl
- IaaS K8S

AI Vision

Steps:
1. Power on
2. Connect camera to POE line
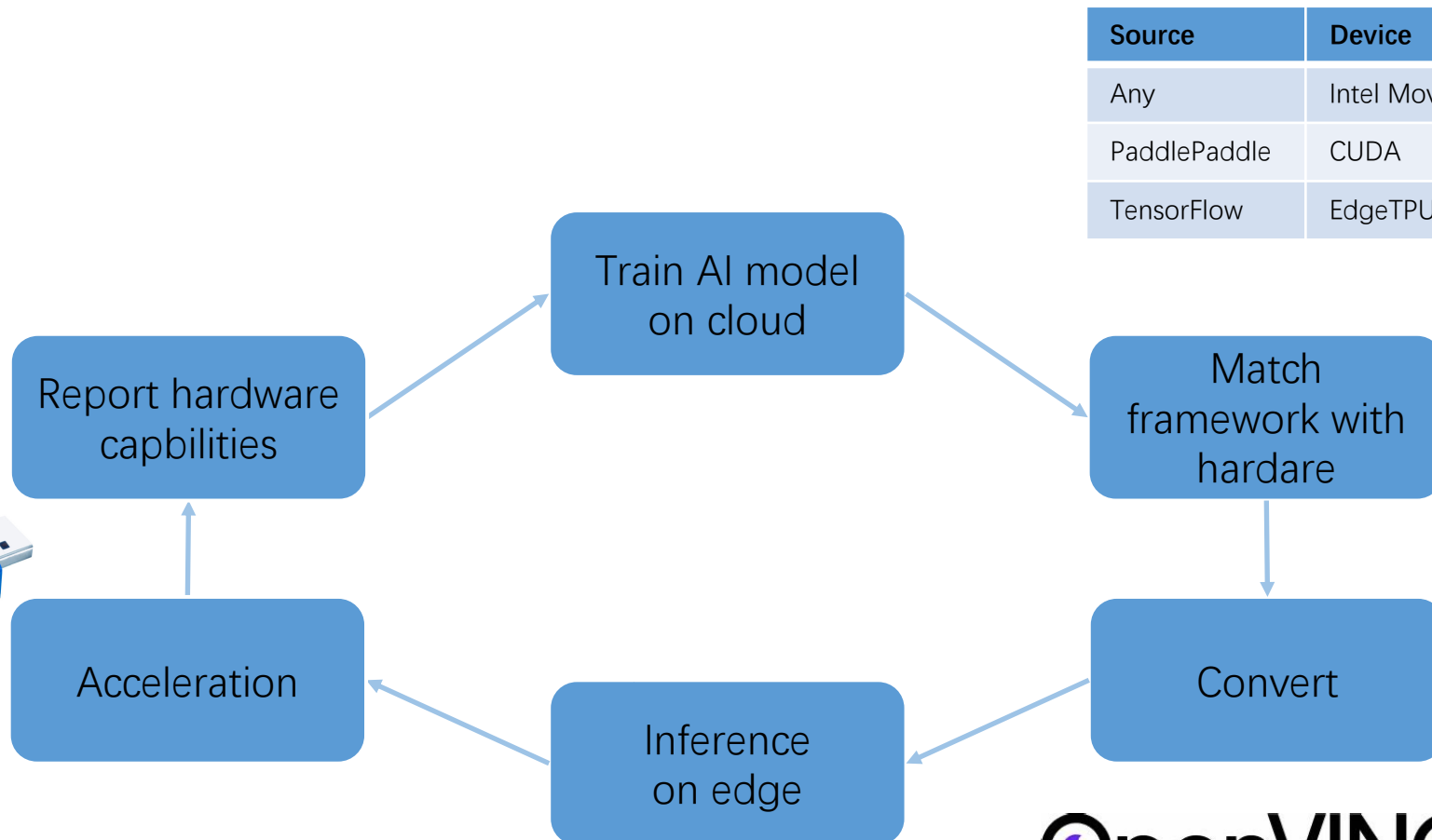3. Connect AI chipset to USB port

Workflow:
1. Load AI model
2. Sampling video as images
3. Do inference, accelerated by AI chipset
4. Save target image
5. Send result to next process node
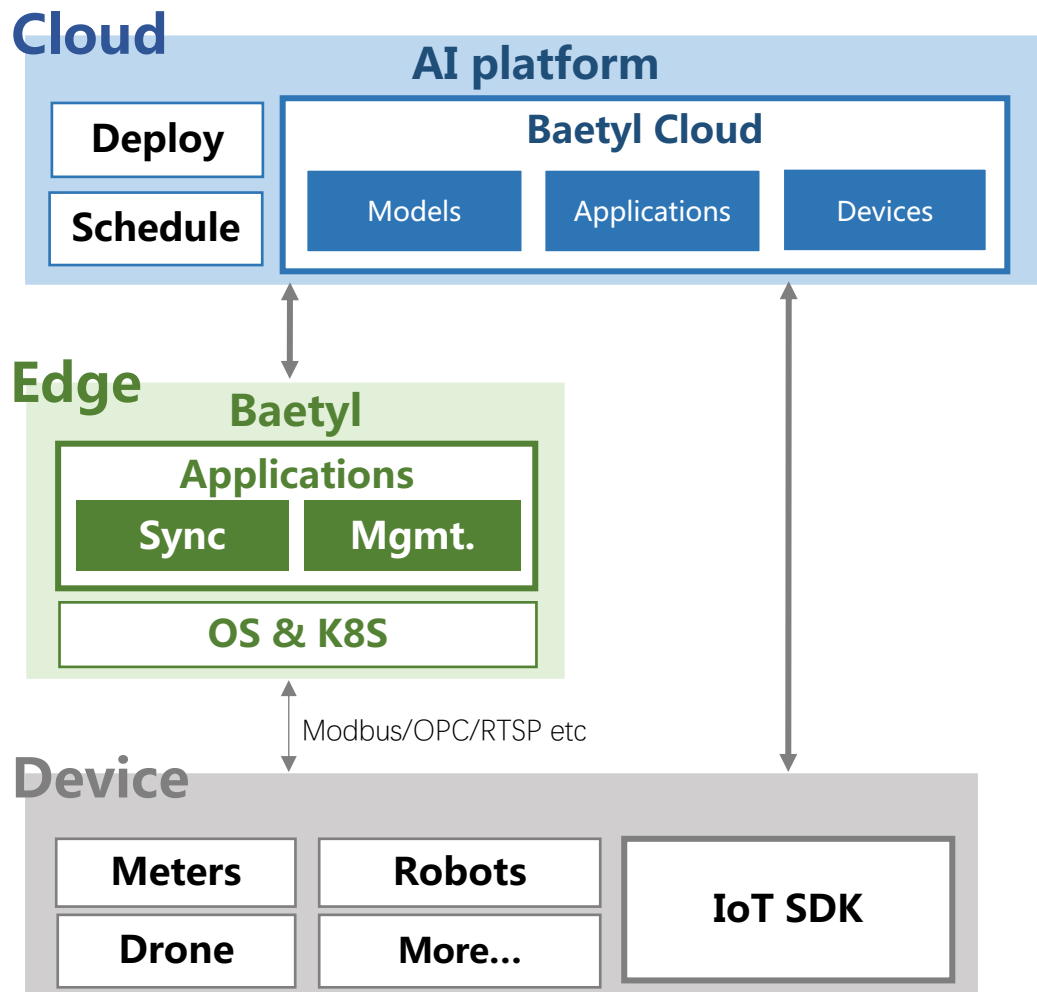6. **Push target image into cloud**

AI auto convert

Source	Device	Target
Any	Intel Movidius	OpenVINO
PaddlePaddle	CUDA	PaddleLite
TensorFlow	EdgeTPU	TF-Lite

Train AI model on cloud

Report hardware capbilities

Match framework with hardare

Acceleration

Convert

Inference on edge

OpenVINO™

Implementing Smart Grid by Baidu IntelliEdge solution

Cloud

AI platform

Deploy

Schedule

Baetyl Cloud

Models	Applications	Devices

Edge

Baetyl

Applications

Sync | **Mgmt.**

OS & K8S

Modbus/OPC/RTSP etc

Device

Meters	Robots	
Drone	More...	**IoT SDK**

The AI platform includes:

- Cloud management suite, responsible for batch distribution and configuration update of AI applications

- Local client, supports deployment on the edge and end, and is responsible for AI application start-stop and state management at the edge layer and end-side

Edge service includes:

- When the local client is deployed on the edge node, the cloud delivers the application to the edge layer, the edge layer collects the terminal data of the device layer, implements prediction and inference on the edge node.

- When the local client is deployed on the terminal device, it is applied to the terminal under the cloud, and prediction and inference are realized on the terminal.
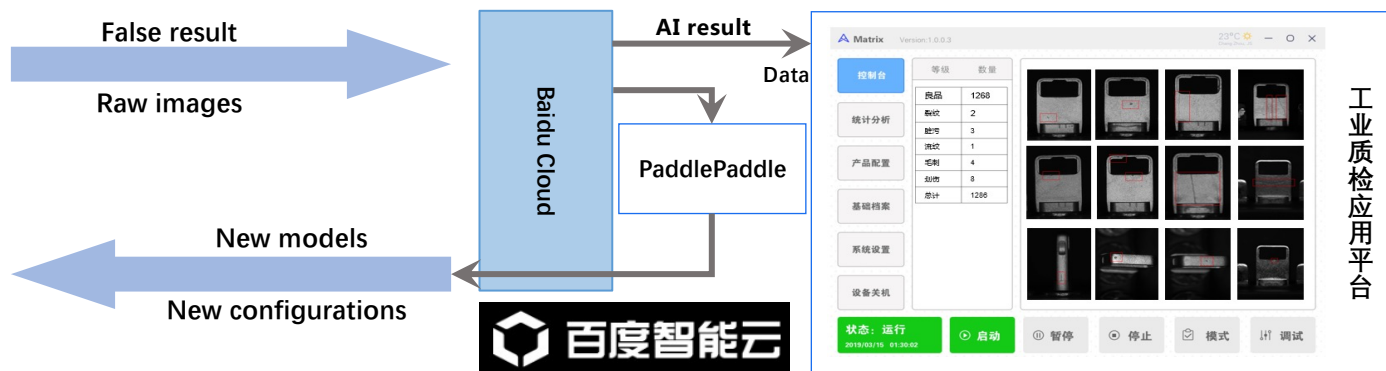
Intelligent quality inspection

Issues

- Model deployment time is long
- Training data collection is complex
- Spare parts quality inspection cloud inference delay affects quality inspection speed
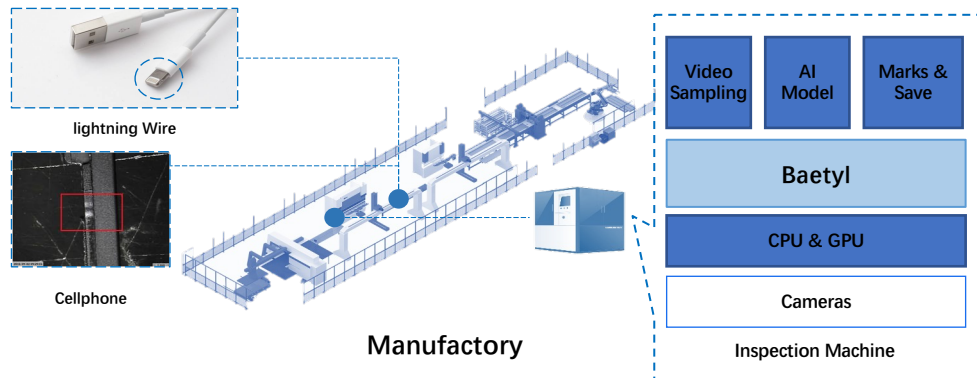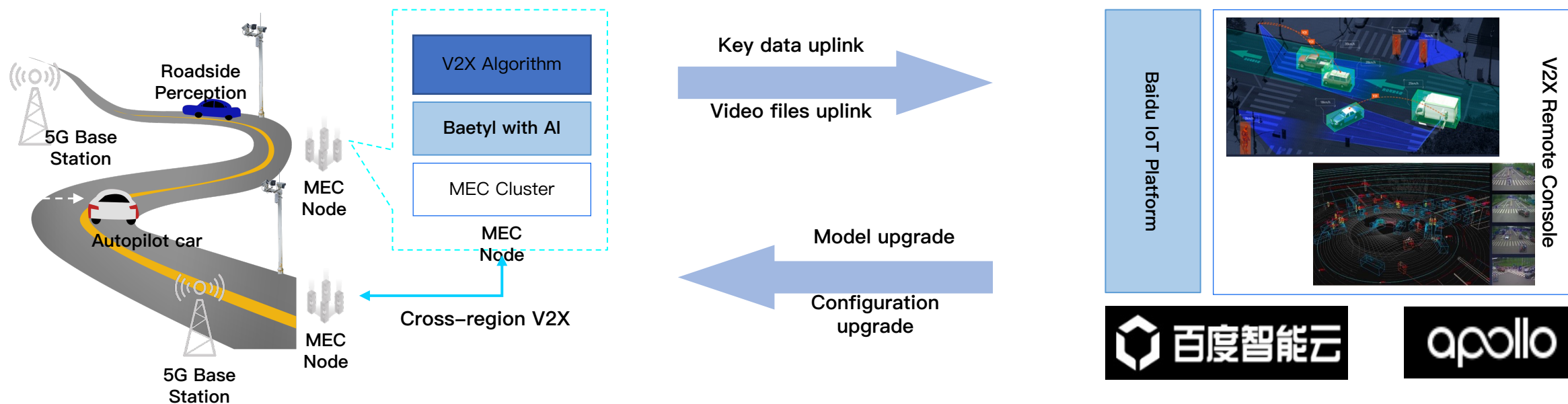
Effects

- Model deployment went from 1 day to 1 minute
- Automatic collection of training data and automatic upload to the cloud
- Edge AI inference realizes edge quality inspection and improves quality inspection efficiency
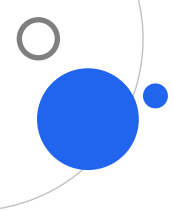
Using Baidu IntelliEdge to achieve vehicle-road collaboration

Roadside
Perception

5G Base
Station

MEC
Node

Autopilot car

Cross–region V2X

5G Base
Station

MEC
Node

V2X Algorithm

Baetyl with AI

MEC Cluster

MEC
Node

Key data uplink

Video files uplink

Model upgrade

Configuration
upgrade

Baidu IoT Platform

V2X Remote Console

百度智能云

apollo

Over 60% reduction in automatic driving costs to help solve cross–regional road
synergy challenges
Global optimization of a wide range of multi–base station nodes

Contact us

https://baetyl.io/
https://github.com/baetyl/baety
https://github.com/baetyl/baetyl-cloud

mailto:baetyl@lists.lfedge.org
lfedge.slack.com/#baetyl

Baidu Cloud IoT
https://iot.baidu.com/

百度智能云AIoT
天工小助手

扫码添加了解更多行业资讯

Thanks

Jun 2022