"Cloud-Native Vehicle" for contemporary digital experience

Keynote @ AGL All Member Meeting 2019 Fall

Hisao Munakata

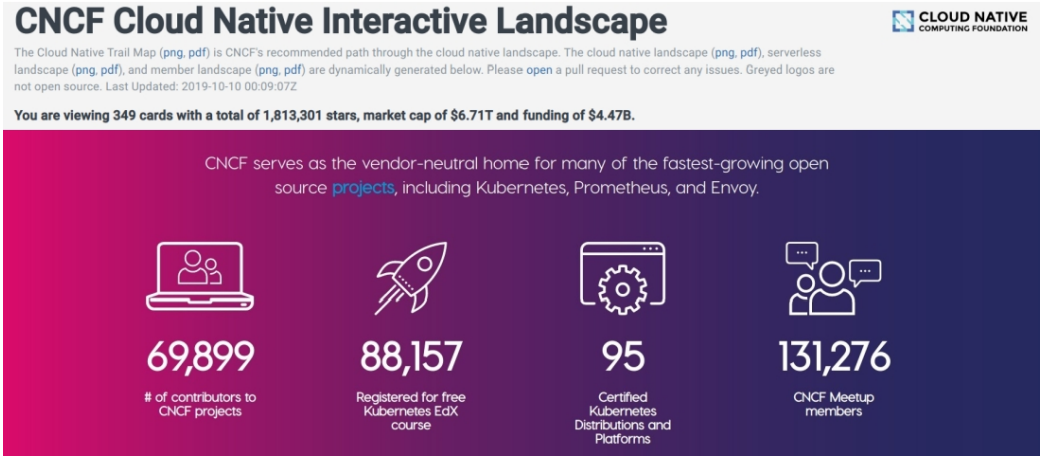
Automotive Grade Linux Advisory board member
Linux Foundation board member

2019-10-22

What is ”Cloud Native” and how it works?

”Cloud Native”concept and its coverage

CNCF project gains huge momentum in Linux Foundation

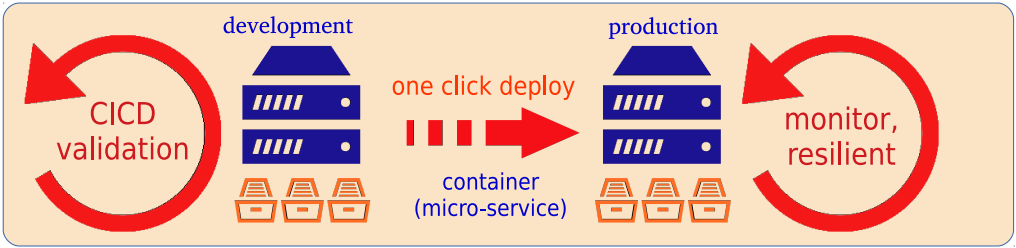


"Cloud Native" definition (is current common sense)

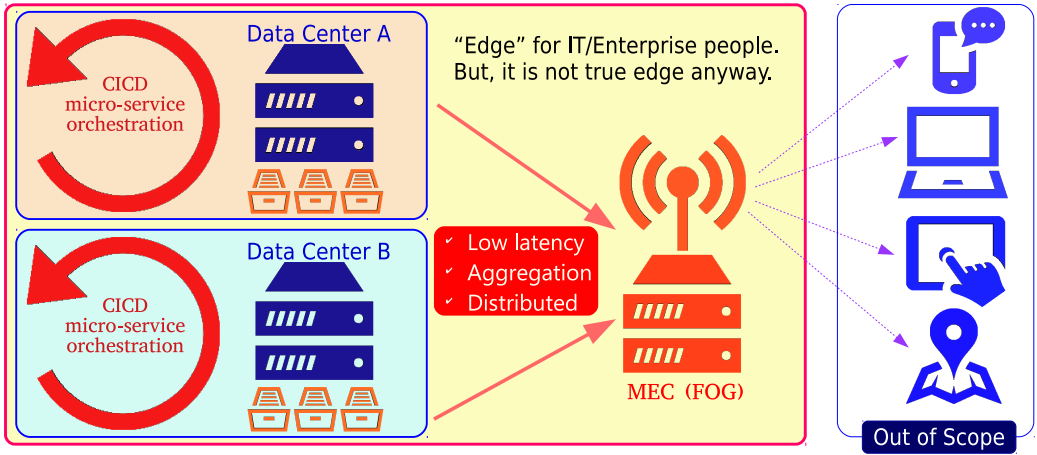
CNCF Cloud Native Definition v1.0

- Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.
- These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil.
- The Cloud Native Computing Foundation seeks to drive adoption of this paradigm by fostering and sustaining an ecosystem of open source, vendor-neutral projects. We democratize state-of-the-art patterns to make these innovations accessible for everyone.

"Cloud Native" scope seems limited to IT/Enterprise (so far)



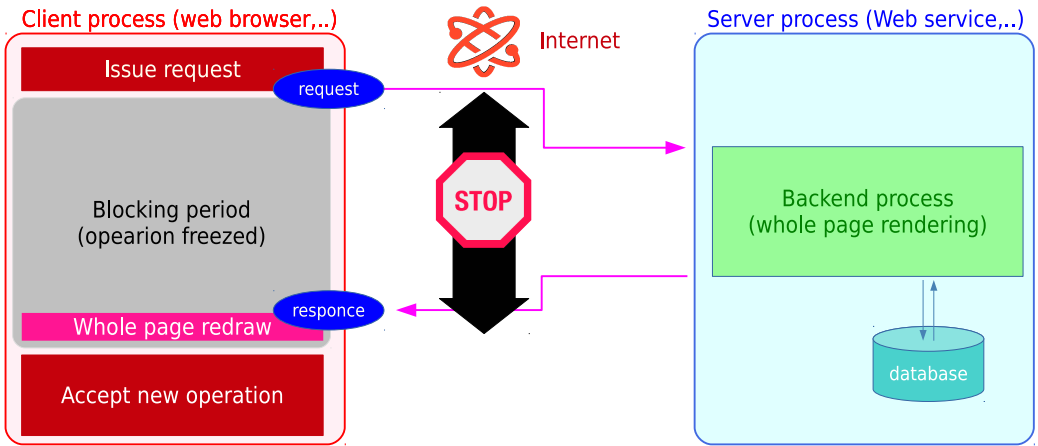
Telecom propose MEC (=Mobile Edge Computing) concept



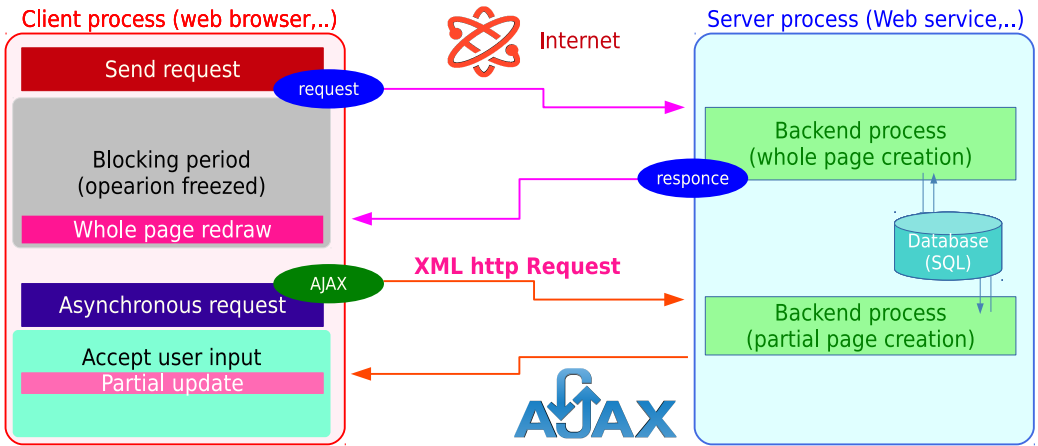
System technology trends

Server side trends (blocking -> asynchronous)

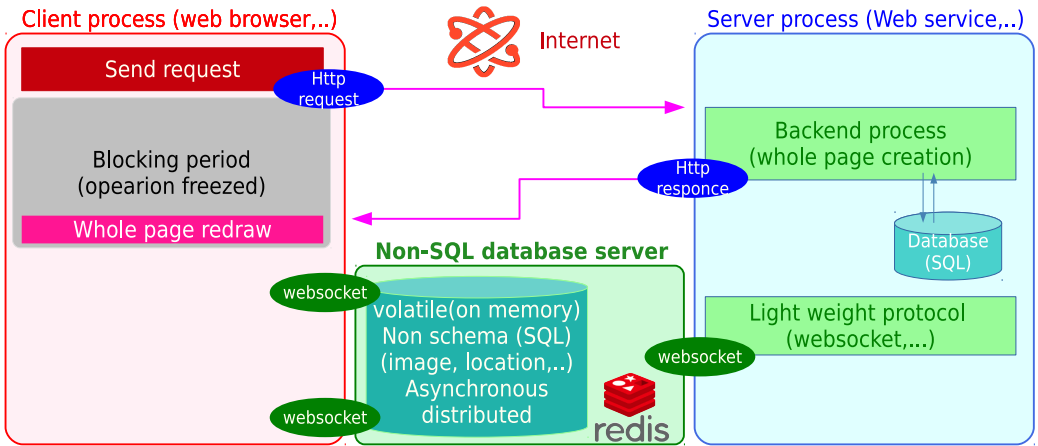
Until around 2010, client_to_server interaction was "blocking"



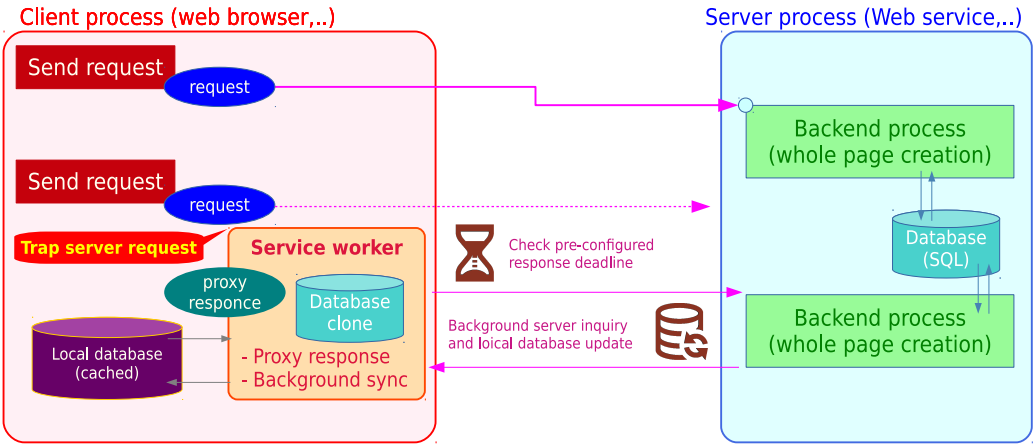
Asynchronous method (1) Asynchronous Javascript And XML



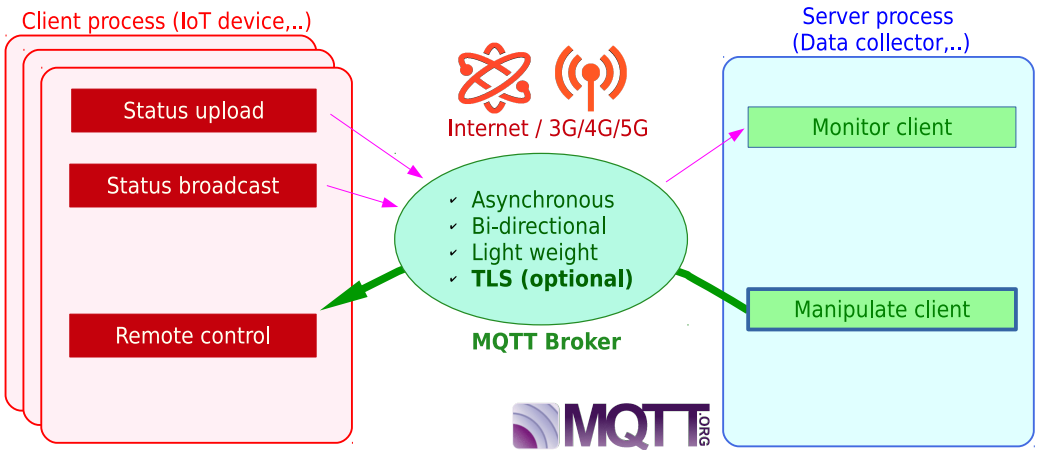
Asynchronous method (2) Non-SQL, on-memory database



Asynchronous method (3) service worker (proxy response)



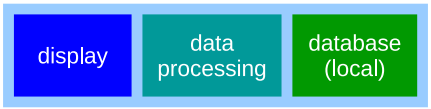
Asynchronous method (4) MQTT (Message Q. Telemetry Transport)



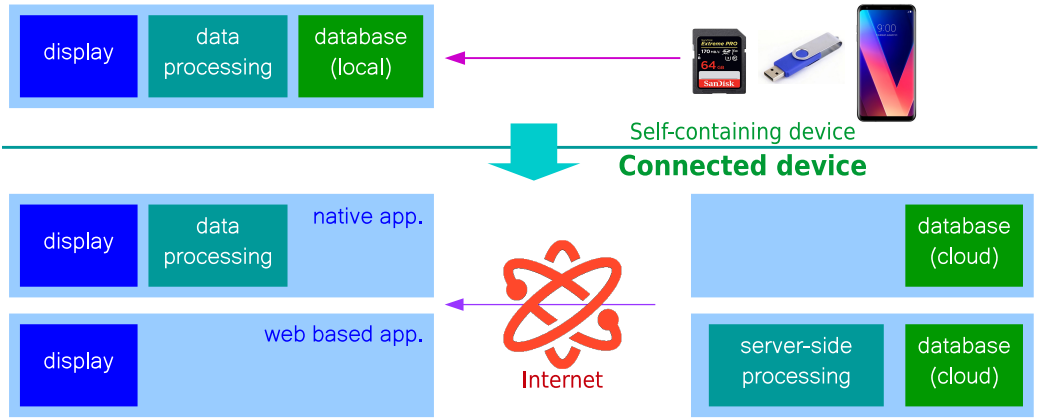
Embedded device : now connected, and then ,..

Embedded trend (1) Self-containing

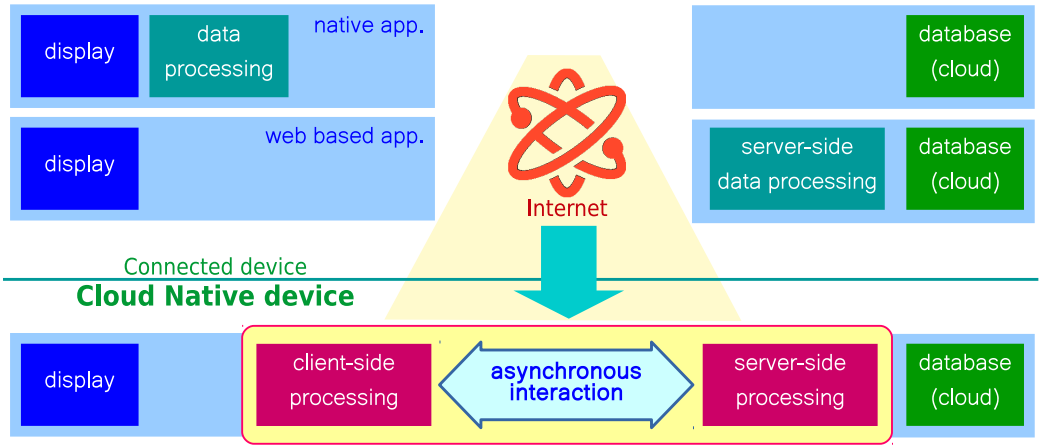
Embedded trend (2) Self-containing + external contents



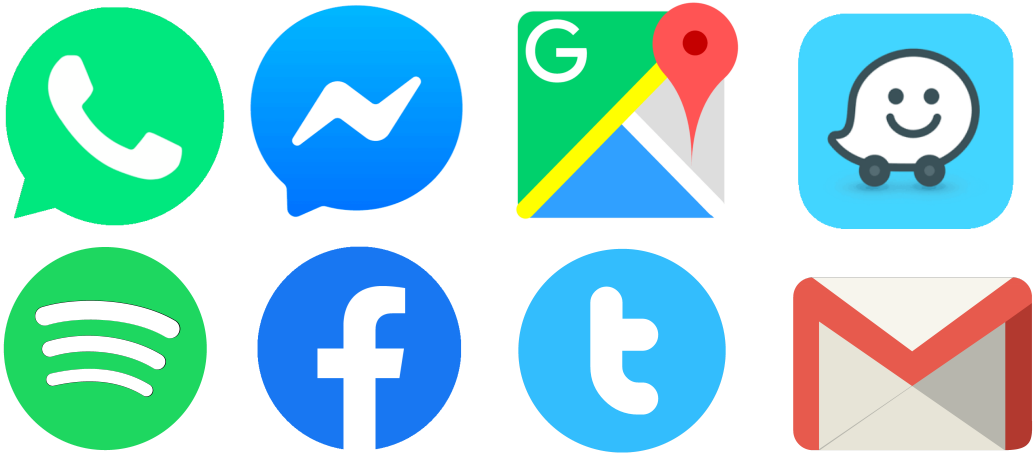
Embedded trend (3) Connected device



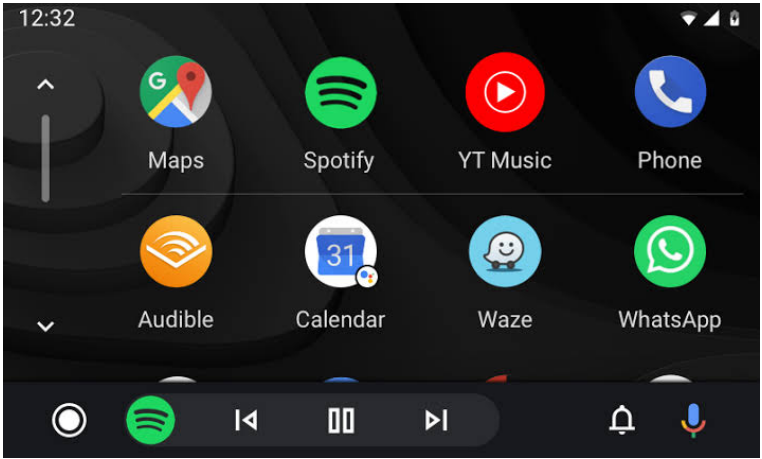
Embedded trend (4) Cloud native device



Modern application come with asynchronous cloud integration



"Android Auto" already include "Cloud Native" capabilities



Android includes

- Seamless apps. (native + cloud)
- Asynchronous cloud connection
- App store
- Data integration (PC, phone and car)

Connected Car != Cloud Native Vehicle

Why smartphone addict so many people?

Vehicle gains smartphone-like HMI, however...

Smartphone-like HMI

- vertical screen:
  - tree-tier split screen (top, main, bottom)
- pinch:
  - map zoom-in / out
- swipe / flick:
  - change apps.
  - flip menu page

"Smartphone-like" should mean "digital experience ready"

User expectation is much more than just HMI design

- Realtime user context understanding
  - geolocation and destination aware guidance
  - schedule lookup
- AI assistant / Agent capability (Siri, Google assistant)
- Social application integration
  - SMS message post, photo upload,..
  - message broadcast, car location share,..
- Software (=Application) update / install

Smartphone already penetrated "digital experience" to vehicle users

Connected vs. Cloud Native, how they differ

Connected = Synchronous (blocking)

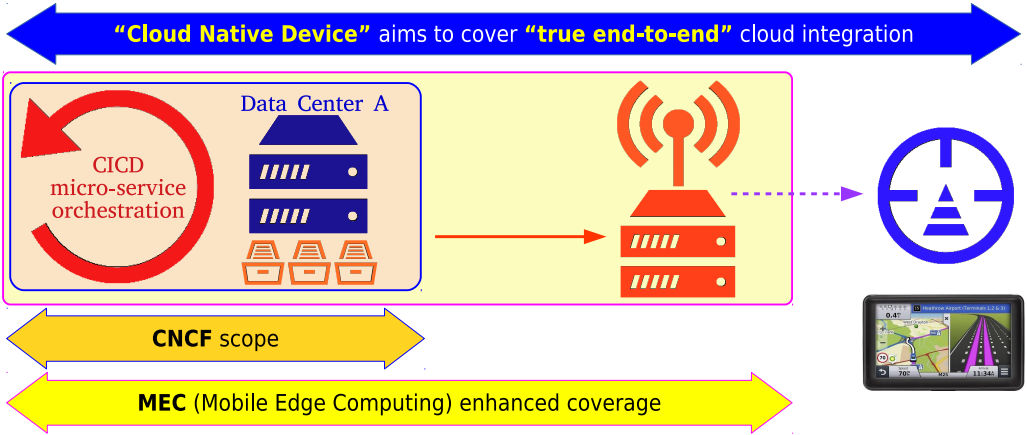
- send a request to the server
- wait for the server response
- client side operation is blocked
- server process user request
- server returns the result
- accept a new user operation

Cloud native = Asynchronous (realtime)

- send a request to the server
- while waiting the responce, still accept a user operation
- server process user request
- new client request by operation
- client returns the result (if possible)
- server returns the result

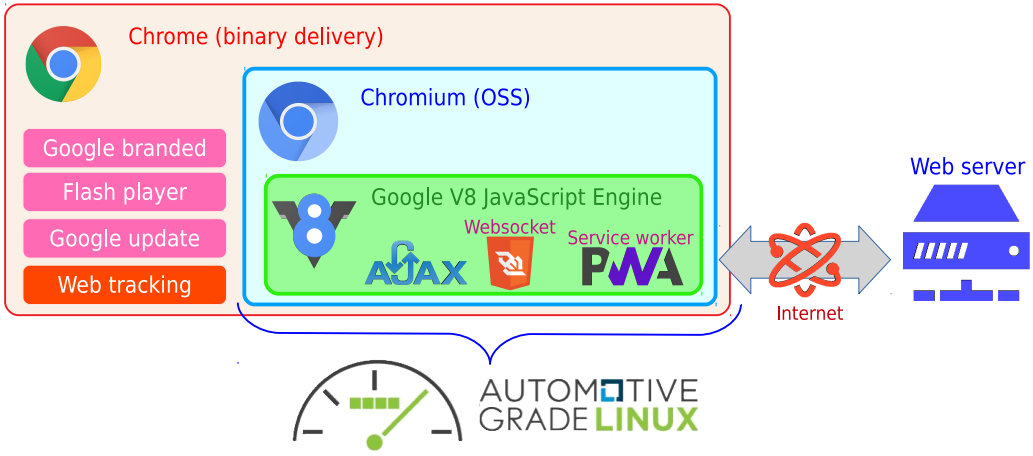
Asynchronous cloud integration realize the "stress-free maneuver"

Brand new "Cloud Native Device" concept



”Could Native device” = utilization of latest web technology

Good news ! : AGL already include a seed of ASYNCHRONOUS



Benefit, benefit and benefit,..seems right direction to go

Benefit for OEM

- **Improve competitiveness**
  (satisfy customer's demand)
- **Increase manageability**
  * Security control
  * Privacy control
  * Server side App. upgrade
- **Thin client (low cost),**
  renovation by SW update
- **Revenue increase**
  by subscription business

Benefit for users

- **Digital experience**
  utilize cloud resources
- **Customize (personalize)**
  by user application install
- **Always new**
  (system never obsolete)
- **Easy to use**
  simple & common method

Benefit for Service

- **Mobility service platform**
  integrate vehicle and cloud
- **Productivity**
  use JS, python language
- **Aggregation**
  realize multi-modal apps.
- **Commonalities**
  one service runs everywhere

The road to "cloud native vehicle"

AGL can be more "cloud native (friendly)" vehicle SW platform

- AGL start integrating some key "asynchronous" pieces,...
  - Chromium browser integration (by Renesas, Igalia)
  - Web Run time (= WRT) support (by LG)
  - Authentification mechanism, Access control mechanism (by IOT.bzh)
  - But, still missing HMI Toolkit, event passing mechanism and more ..
- Now we should not invent unique solution, alternatively let's learn from IT/Enterprise forks who develop asynchronous cloud integration
  - Embedded developer should learn from cloud-native guys (or recruit them)
  - We need work with them to add automotive-specific demands like safety, robustness,...

Conclusion

key messages today

- Just "connected" is not enough to satisfy contemporary car users who already fully utilize smartphone capabilities.
- Alternatively, today I proposed "cloud native vehicle" concept that aims to provide true digital experience to the drivers and passengers.
- To realize cloud native digital experience, we need to adopt latest cloud/edge coordination technology (called realtime-web infrastructure) that started roughly 10 years ago in the IT/Enterprise industry.
- The Linux Foundation is the perfect right place to make vehicle and cloud technology synergy (lead by the AGL and CNCF collaboration).