



車載マルチメディアにおけるLXC コンテナ適用例

木村 秀人

プロフィール



木村 秀人

- 2003.04 富士通テン(現デンソーテン)入社
カーナビ/オーディオのシステムドライバソフト担当
8bit-32bitマイコン(NEC/Renesas/Fujitsu/Panasonic...)
- 2009.04 周辺監視3Dカメラ開発
システムアーキテクチャ/3D Graphics設計
- 2013.04 17年向けカーナビ/オーディオ向けLinux PF開発
- 2017.04 21年向けカーナビ/オーディオ向けLinux PF開発
Udacity 自動運転 Nono degree取得(一期生)
- 2020.04 24年向けカーナビ/オーディオ向けLinux PF開発

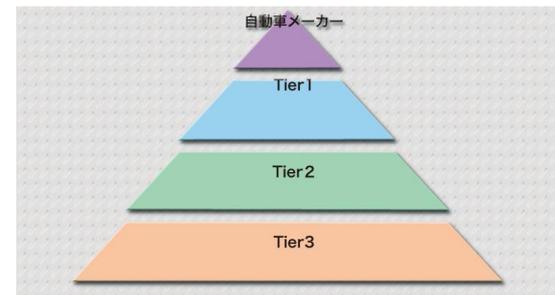
趣味:ドラム演奏、DIY

第一部 車載マルチメディア(IVI)とは？

車載マルチメディア(IVI)をまるっと説明

- ・IVI (In-Vehicle Infotainment)
- ・いわゆるカーナビ・カーオーディオのこと
- ・使うか使わないかは別にしていっぱい機能あり
(クラウドナビ, 音声認識, スマホ連携, etc)
- ・車メーカーの固有通信。それぞれガラパゴス諸島。
(CAN, LIN, FlexRay, Ethernet, AVC-LAN, etc)
- ・高級車両はメーカーオプション(ライン装着)で販売

前提



- 本日はお話するのはライン装着品のソフト開発
- 開発期間は、企画～販売まで約3年。
- 保証期間は車両と同じ最低10年。
- Tier1という特殊な制度がある。

ティア1とは(Tier1)とは、一次請けという意味があり、自動車業界では完成車メーカーに直接部品を供給するメーカーのことを指します。

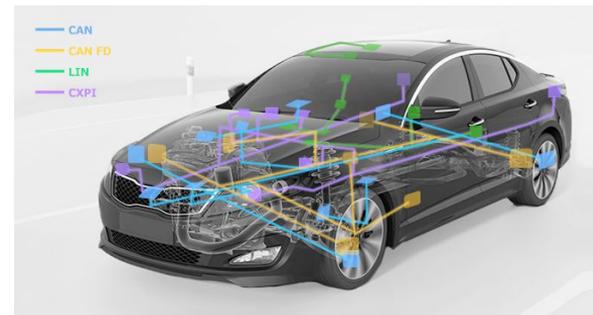


特徴(1)

- とにかく開発が長い。石の上にも3年。
 - 海外では最初に企画した人間はとっくやめてる企画意図をTier1が語ることもしばしば。
 - 日本の会社でさえ企画担当が変わりリセットもしばしば。
- とにかく保守も長い。車のライフサイクルとセット。
 - メジャーチェンジ(4年)、マイナーチェンジ(2年)
 - 新機能が他社からでると機能追加される。

特徴(2)

- (いうほど)標準化が進んでいない
 - コストや車の搭載位置でECU構成がコロコロ変わる。
 - 現在ECUは30個以上、自動運転/EV化で100という試算も。
 - もはや巨大なロボットと同じ。
 - 分散システムが割と早期から確立しているため、メッセージパッシングの思想はよくできている。
しかし、通信仕様が車メーカー(OEM)ごとで異なるため、標準化にはほど遠い





特徴(3)

- (いうほど)サーバ接続(コネクティッド)できない
 - 安定した通信は海外では不可能。
 - 日本はともかくとして、
走行し続ける車にとってよい接続環境ではない。
 - FOTAなんてもってのほか。
 - サブスクにお金を払える人は一部の富裕層だけ。

※ただし、アメリカ市場の富裕層は 2年間で新車を乗り換えるので、イーロン・マスクのように夢を見させてくれれば、多少の不便は目をつぶるようです。

特徴(4)



- 車両設計に合わせた短期間でのイテレーション開発
 - 元祖アジャイル開発の「張り子の虎」設計
 - 半年～1年で試作機を複数回作ることで完成度をあげていく。仕様はもちろんフワフワ。
 - 結局、数か月でモノを作るため、人の取り合い。
 - 最初の一年はハード設計のため、バリバリのウォータフォール開発
 - 完成度が上がるごとにイテレーション開発に移行

But、日本の産業には合ってる！！

- **日本が自動車大国でいられる理由**
 - **長期間の部品供給が必要なので、継続的な工場の設備投資が必要**
 - **規格・安全に対しての国際ルールが厳しい。**
 - **10年保証するための人材の継続雇用が可能。**
 - **多品種少量生産。多機能工。**
 - **全世界で売れる。ローカライゼーションが必要。**
 - **価格が下落しにくい。**

自動運転ブームはなぜ減速したのか？

- 2016年 自動運転元年と呼ばれたが・・・。
 - この立役者はもちろんGoogleの自動運転カー
 - シリコンバレーが起点となってAIエンジニアの奪い合い
 - しかし、シリコンバレーのベンチャーは一年で結果が出ないと事業継続できない。
 - 長期開発/保証を超えて生き残ってるのはテスラだけ。時価総額は世界一だが、販売台数は数10万台。



EVブームはやってくるのか？

- 2021年 EV元年と呼ばれるらしいが・・・。
 - 寒冷地ではバッテリーが弱まる。
 - 充電ステーションの供給問題。
 - 環境には全くやさしくない。
 - もちろんコスト高。本体価格も上がる。
 - 問題が山積みのまま、各国EV移行宣言している。
 - 産業は政治が作るの現時点ではわからない。

第一部 車載マルチメディア(IVI)とは？

まとめると、

- ・開発3年、保守4年、保証10年
- ・OEM毎に車両通信方式が異なる
- ・サーバ接続性が決して良くない
- ・車両(ハード)と合わせたイテレーション開発

日本の強みも活かせる！

**長期開発／生産／保証できる企業体力、人材雇用
多品種、ローカライゼーション、機能安全への対応**

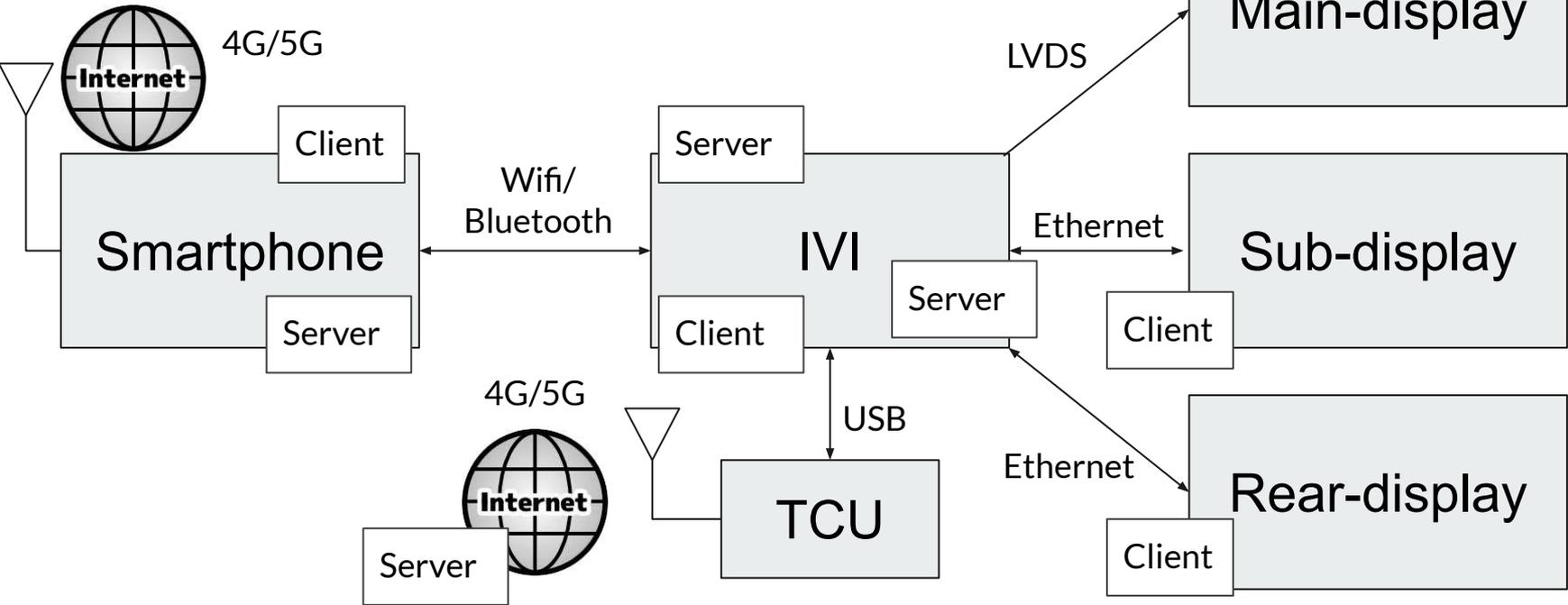
第二部 車載IVIの中を見てみよう

車載IVIの中ってどうなってるの？

- **実はOSSが70%以上、年々増加中**
 - linux kernel, wayland/weston, gstreamer, alsa
 - sqlite, free type font
 - chromium, bind, openssl, libnss, curl, grpc
 - http server, ftp server, dhcp server/client, websocket, avahi...
 - **特に近年はサーバ用途が増加。表示機器でありながらIVIの中にはサーバがいくつも立っている！！**

TCU:テレマティクス制御ユニット・通信モジュールのこと

周辺機器との連携例





車載IVIの目下課題

- サーバ、スマホ連携に機能アップデートが集中
 - サーバプッシュ
 - クラウドナビ
 - クラウド音声認識
 - 車両情報(カメラ、GPS)のセンターアップロード
 - スマホを使った車両設定、個人情報連携
 - 最新のOSSを使って、
サーバ、スマホアップデートへの追従が必須



アップデートへの課題

- **脆弱性対応も含めアップデートのハードル高し**
 - 人気があるOSSは脆弱性も多い
 - LTSがないOSSも当然ある
 - 定期アップデートに難点あり
 - ※接続/コストが課題、そもそもTCUを契約していないユーザが大多数
 - **価値を生み出すコアコンテンツに乏しい**
 - ※クルマでサブスクしている人が圧倒的に少ない。
 - **価値を生み出しながら、定期アップデートする方策が必要**

クルマで価値あるコンテンツとは？

- **安全で快適なドライブを支援する**

- **クルマも一つのセンサー部品、IoT機器**

- ※車内外カメラ、マイク、自車位置情報(GPS+Gyro+Navi)

- **AIによるドライバ状態検知、走行状態検知**

- ※保険の査定にも利用可能

- **道路情報のクラウドシェア**

- **自動運転支援(ADAS)**

- **地図との情報連携**

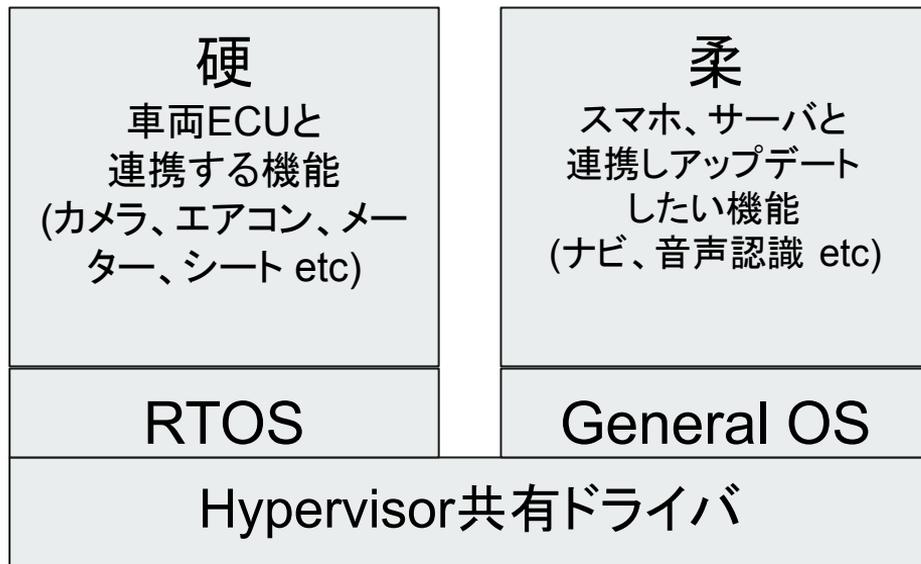
車業界外のほうが使い方が
上手ではないか？
高く売れるのではないか？



本日の重要ポイント

- **車載で仮想化が必要な理由**
 - **柔硬分離**
 - ※硬(機能安全、車両操作)と柔(エンタテイメント)の両立
 - ※定期アップデートに耐えられる構造
 - **ハードウェア分離**
 - ※開発期間3年の内、1年はハード/ドライバ設計
 - **過去資産利用、別OSとの共存**
 - ※RTOS、Android Auto OSなど

車載IVIでの仮想化と課題

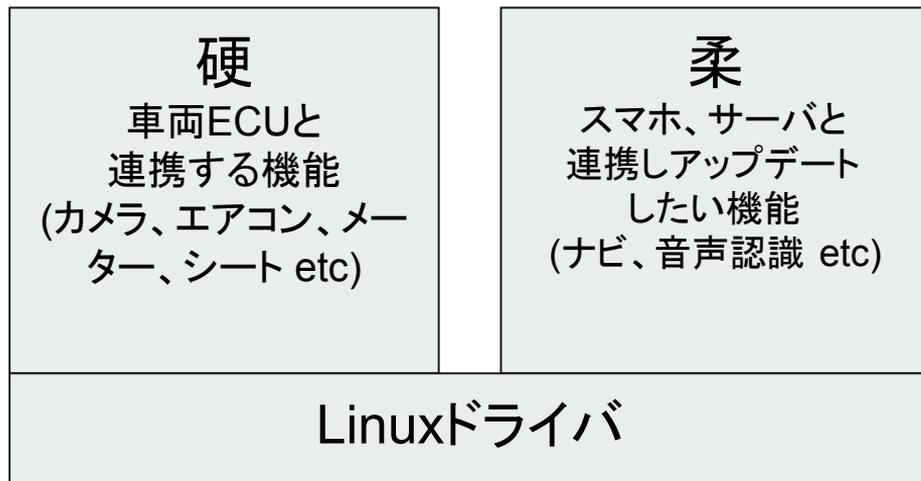


課題

結局、共有が必要なドライバが非常に多い

結局、硬領域を優先する事になるため、Linux以外のOSでドライバの再設計が必要
開発コストが跳ね上がる

コンテナにしてみたら？



開発レスでお手軽に柔硬分離可能！

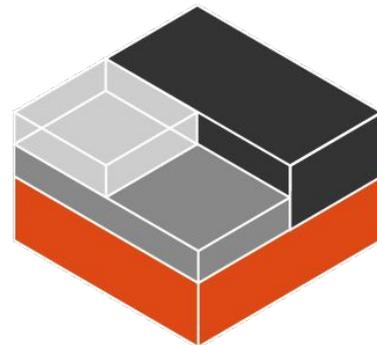
PROS

Kernelから見たら
ただのプロセスなので、
そもそもドライバを共有しなく
ても同時動作可

CONS

機能安全には課題が残る
Android Auto OSは
ソフト構成が異なるため、
少し改造が必要

LXCコンテナとは？



(wikipedia) <https://ja.wikipedia.org/wiki/LXC>

LXC(英語: Linux Containers)は、1つのLinuxカーネルを実行しているコントロールホスト上で、複数の隔離されたLinuxシステム(コンテナ)を走らせる、OSレベル仮想化のソフトウェアである。

- **linux kernelのcgroup, namespace等を利用してコンテナを実現！**
シンプルかつ軽量
- **Dockerと何が違うの？**
Dockerはアプリケーションを実行するための依存関係ライブラリの構築の容易さを重要視している。
Dockerの最大の強みはDocker Hubによるイメージ共有とバージョンニング。
車載IVIでは今のところ必要ではない。



LXCコンテナ触ってみたい？

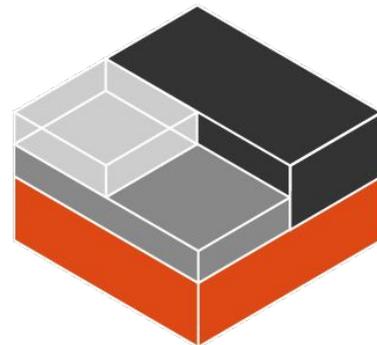
ubuntuにはapt-getで秒で入る。下記がおすすめ。

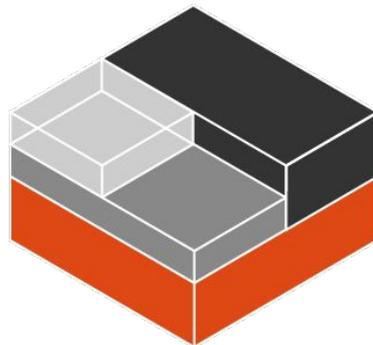
LXCで学ぶコンテナ入門 - 軽量仮想化環境を実現する技術
https://gihyo.jp/admin/serial/01/linux_containers/0001

組み込みLinuxの人には、AGL IC-EGで公開中。
R-CAR H3/M3 SKでお試しできます。

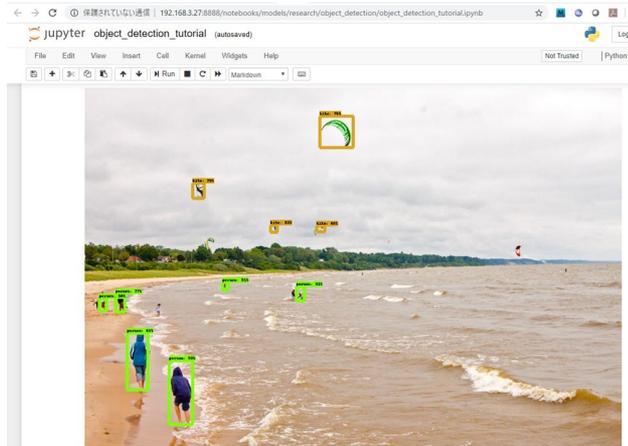
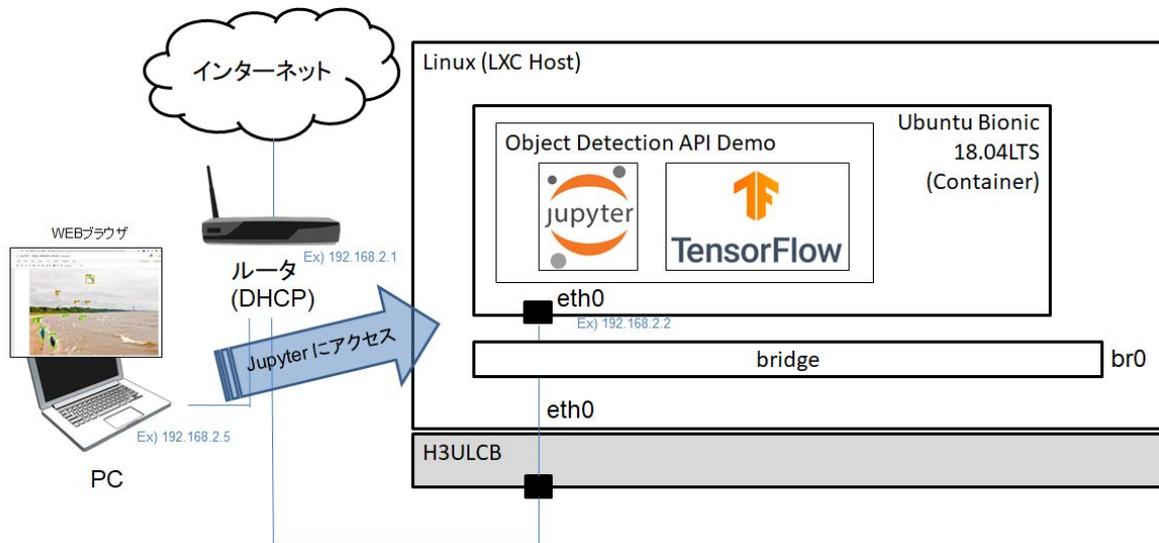
コンテナ環境の作り方

<https://github.com/agl-ic-eg/document/blob/master/README-jp.md>





LXCコンテナ on R-CARで遊んでみた



第二部 車載IVIの中を見てみよう

まとめると、

- ・車載IVIの中にはOSSがいっぱい
- ・近年サーバ、スマホ用途のパッケージが増加中
- ・機能アップデートは必須、しかしビジネス課題あり
- ・アップデートするためには柔硬分離が必要
Hypervisorではコスト高になってしまう
- ・LXCコンテナを使って、お手軽に柔硬分離！
- ・yoctoガチガチのLinuxにUbuntuだって載っちゃう

第三部 車載IVIでのLXC適用例

車載開発での悩み・・・

- (1) 開発期間が長すぎて、企画当初のUIデザインが陳腐化してしまう
- (2) 走行時のネットワーク接続環境に難があり、操作レスポンスが非常に悪い
- (3) せっかく最新のAIを使って学習したモデルを持っているのに、ユーザのうれしさにつながらない

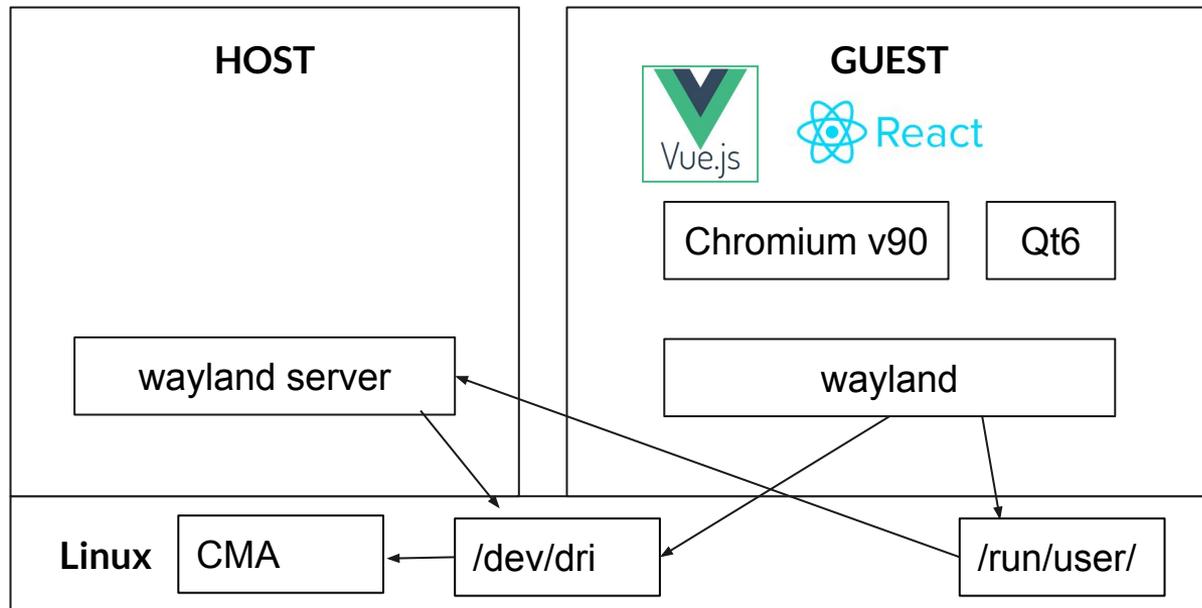


**(1)開発期間が長すぎて
企画当初のUIデザインが陳腐化してしまう**

**解決策:LXC Guestに
最新のUI Framework/ブラウザを導入**

解決策:LXC Guestに 最新のUI Framework/ブラウザを導入

- Wayland protocolのUnix Domain Socketを共有(/run/user)
- Graphicsドライバ(/dev/dri)を共有する



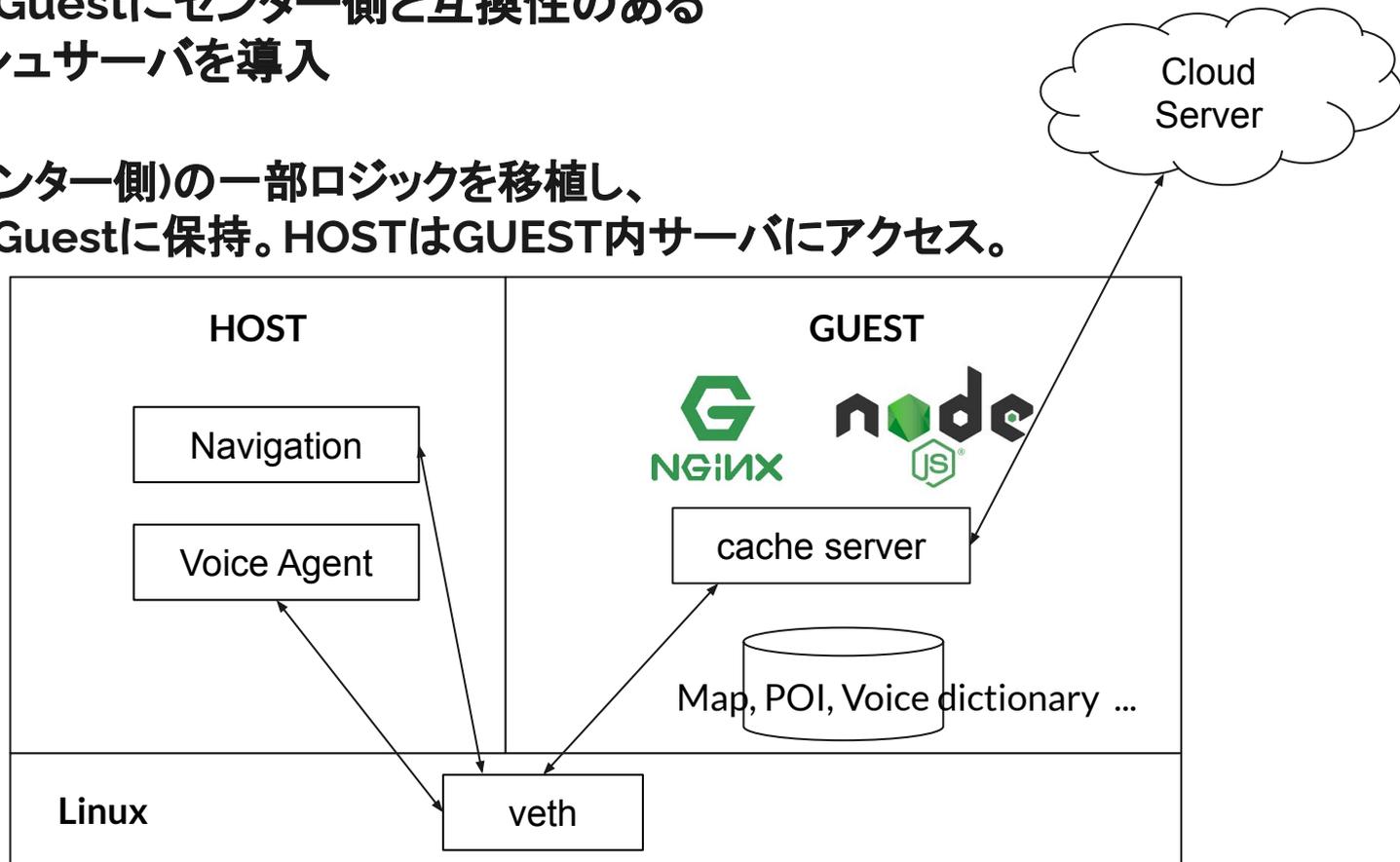


**(2)走行時のネットワーク接続環境に難があり、
操作レスポンスが非常に悪い**

**解決策:LXC Guestにセンター側と互換性のある
キャッシュサーバを導入**

解決策:LXC Guestにセンター側と互換性のある キャッシュサーバを導入

- Cloud Server(センター側)の一部ロジックを移植し、
キャッシュをLXC Guestに保持。HOSTはGUEST内サーバにアクセス。



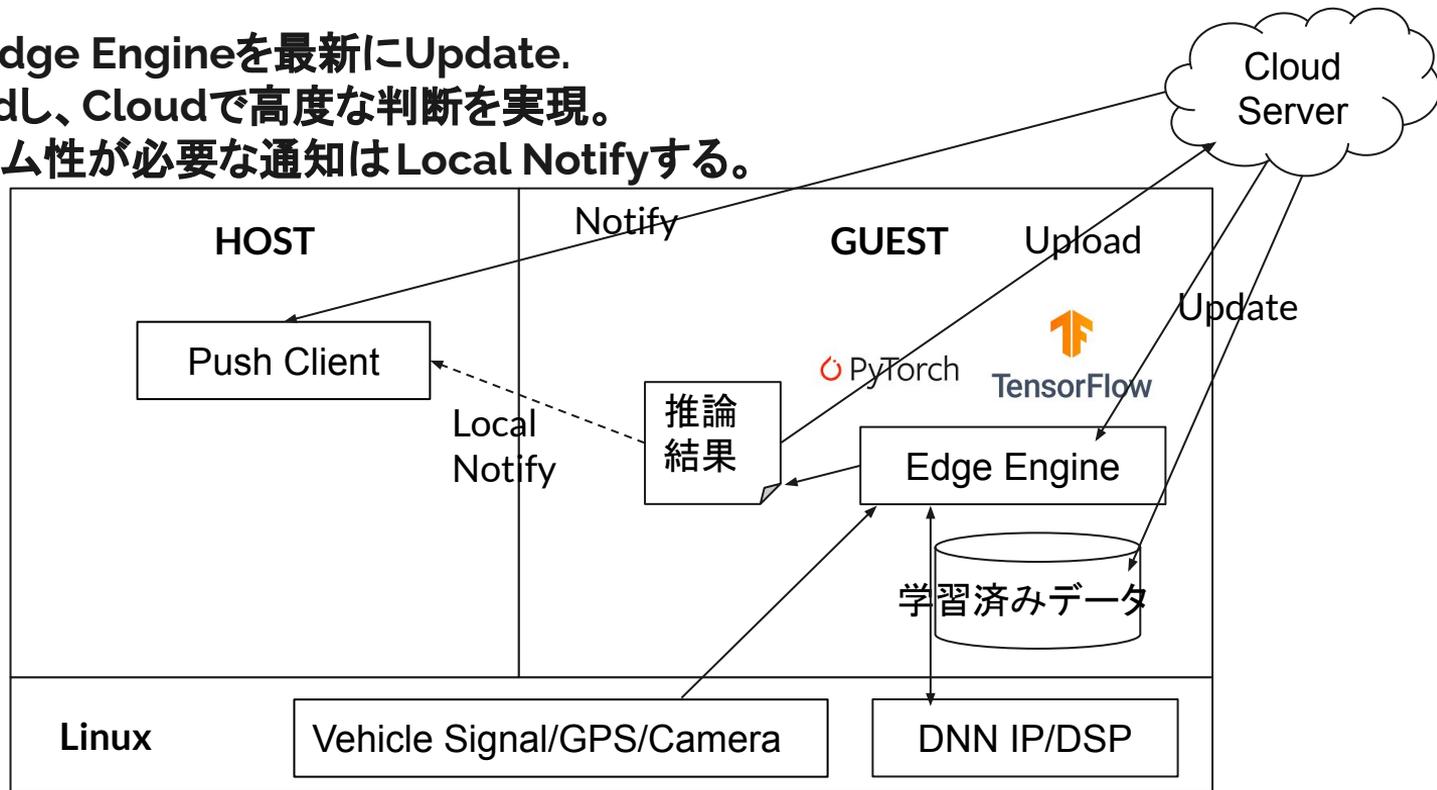


**(3)せっかく最新のAIを使って学習したモデルを
持っているのに、ユーザのうれしさにつながらない**

**解決策:LXC Guestにセンター側と互換性のある
Edge Engineを搭載。
生データではなく推論した結果をアップ。**

解決策: LXC Guestにセンター側と互換性のある Edge Engineを搭載。生データではなく推論した結果をアップ。

- 学習済みデータとEdge Engineを最新にUpdate。
推論結果をUploadし、Cloudで高度な判断を実現。
あるいはリアルタイム性が必要な通知はLocal Notifyする。



第三部及び全体の結論

コンテナでIVIは新たな機器に生まれ変わる！

- **Android/iOS進化にUIで追従するのはナンセンス
そもそもユーザ数も開発規模も異なる**
- **Linuxの強みを活かすには、
新しいOSSの投入しやすさがカギ**
- **素のLinuxに近い開発キット(SDK)を作ること
開発者へのハードルを下げる。**
- **車業界外とのコラボ推進。**

おわりに



ご清聴ありがとうございました

**IT技術の進化は日進月歩、
まだまだ我々にやれることはあるはず**

**日本人ならではの強みを
追及していきましょう！**