



コンテンツ屋が昔つくった PGW OSSの紹介

2022/07

株式会社ミクシィ 吉野純平

- 吉野純平
- 株式会社ミクシィ 開発本部 本部長
- スキル守備範囲
 - ネットワーク関連 (IPルーティング、L2、MPLS (L3VPN,LDP) 、伝送初心者)
 - モバイル関連(S5,Gp)
 - 映像関連(SMPTE2110,SMPTE2022,配信,WebRTC初心者, PTP 2059-2)
 - サーバ運用
 - 電子回路 初心者
 - 3DCAD 初心者
 - コントリビューション先 libgtpnl, tungsten fabric(旧open contrail時代), keepalived, wireshark

• 過去

- モバイル関連の事業検討がありました
- PGW/GGSN作りしました。

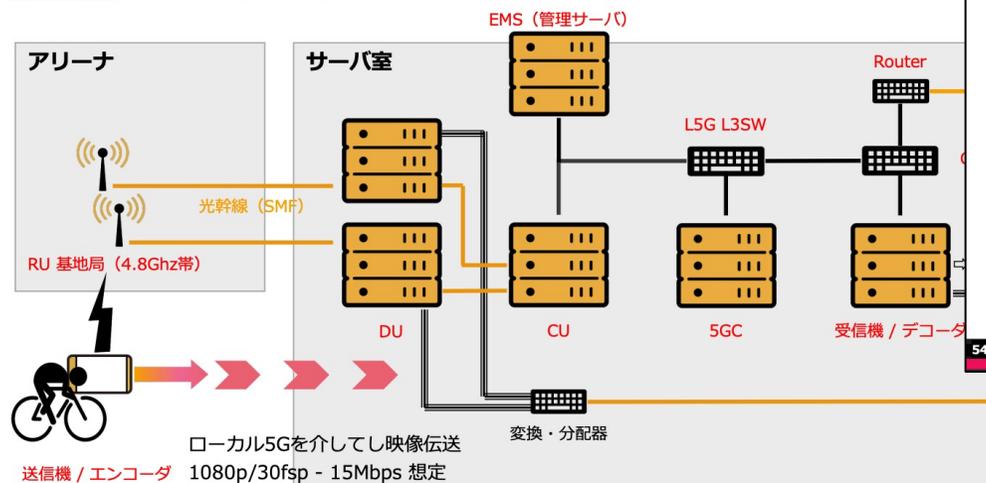
• 現在

- ローカル5Gの競技での映像配信活用
 - 自転車につけるカメラ端末を必死に開発
 - JANOG50で製造機器の写真など公開予定
- TIPSTAR DOME CHIBA内で免許をとって基地局運用

- Day3 2022年7月15日(金) 10:45~11:30(45分)
- <https://www.janog.gr.jp/meeting/janog50/local5g/>

ローカル5G 車載カメラ ネットワーク構成

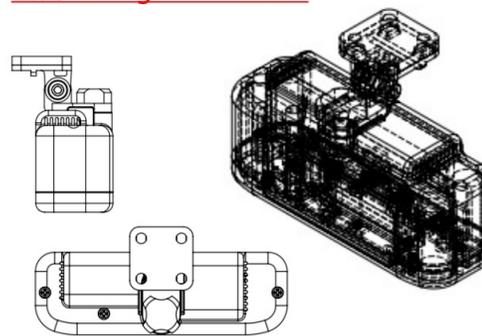
ローカル5G 基地局を無線設備のドーム内高所に設置。サーバ室でSDIに変換後にTV室/ミクシィ社製の送信機・受信サーバを利用してIP映像伝送



あ。ハード製造してます。車載カメラ筐体開発

8 とにかく中身の隙間をなくす。結果、こうなる予定です

型番 : m5gvc0004-250



RaspberryPi Zero + RaspiCam + 5G dongle + ケーブル
スーパーキャパシタ + 中継基板 (自作回路)



3Dプリントしたモック機
商用機は樹脂系素材で製造



正面から m5gvc0004-250 と iphone12mini の比較

※2022年7月現在。開発中のモノです

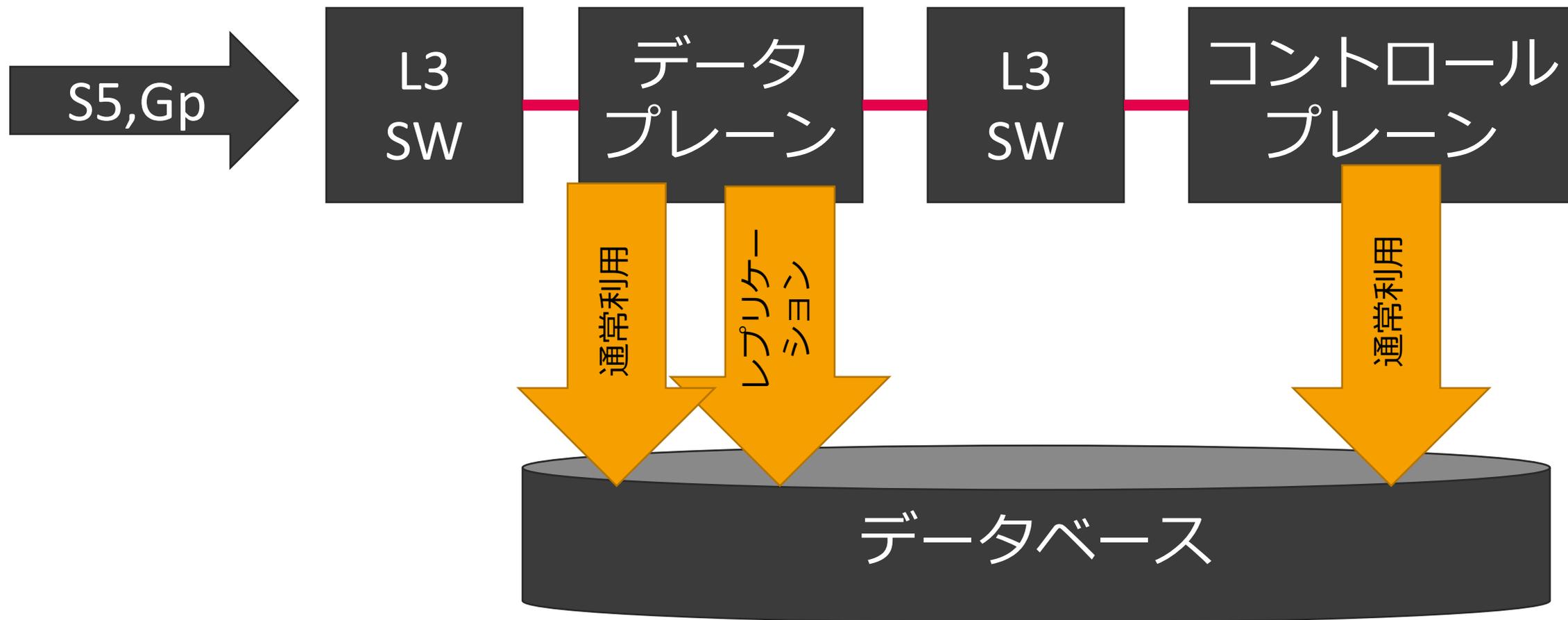
- 迂回が自動的にされる
 - メンテナンス時に迂回が心理的に楽
 - 障害時の迂回も考慮
- 単一障害でコントロールプレーンがばたつかない感じが良い
- データプレーンもコントロールプレーンもCIできる

かなり紆余曲折しながらですが、上記に近づくように模索しました。
socket + kernel module + tap/netmapなどなど

- <https://github.com/mixigroup/mixi-pgw>
- S5,Gpに対応
- コントロールプレーンの各処理にOptionalであるPrivate Extensionを付与
- データベースのレプリケーション機能を活用して、データプレーンをミラー
- レプリケーションは遅延があるので、Private Extensionで即書き換え
- 冗長系にはレプリケーションのみで対応
- データベースを使ってコントロールプレーンとデータプレーンの分離

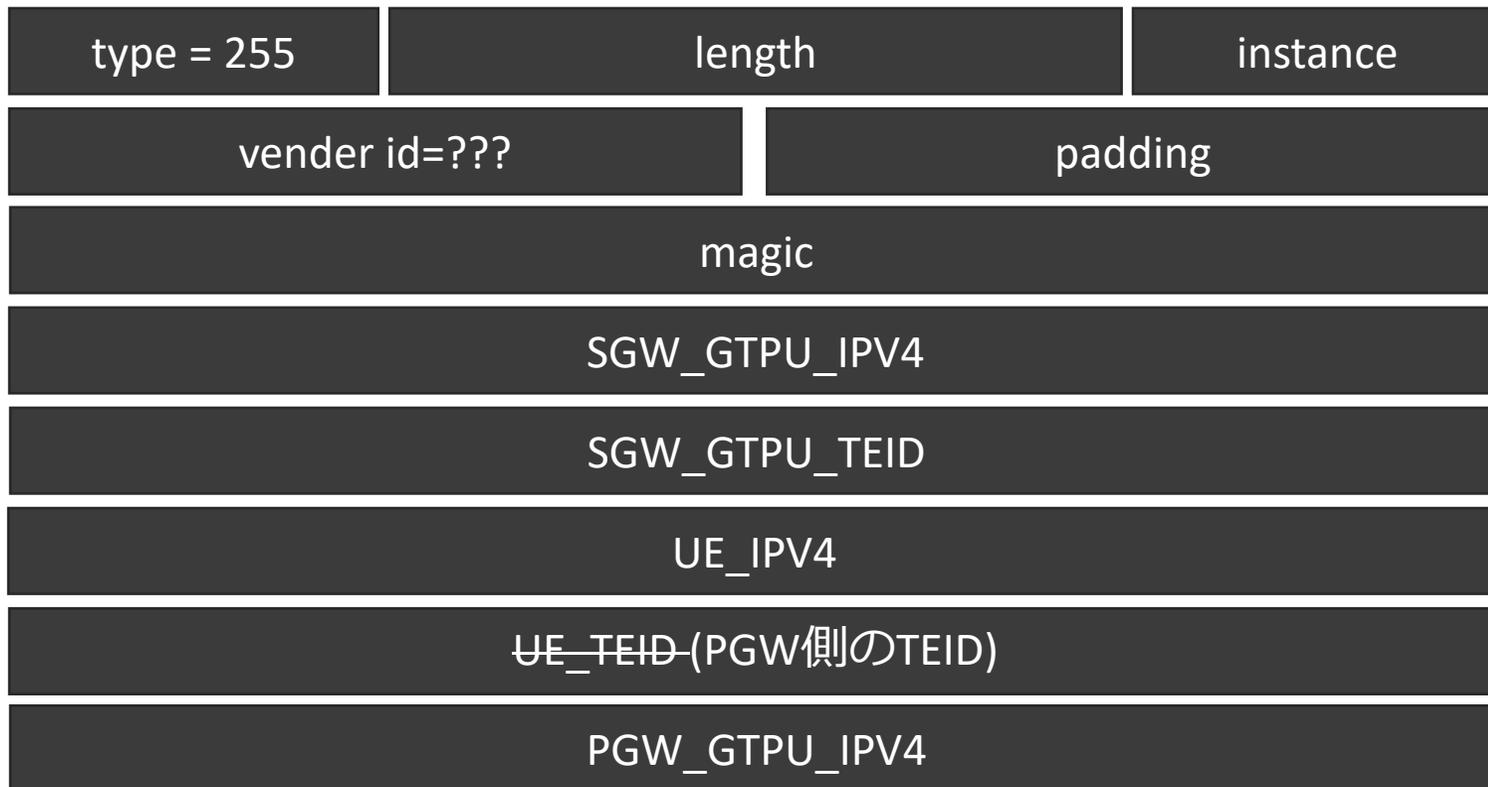
- スナップショットと差分伝達 (binlog)
 - 以下の2つがあればレプリケーションができる
 - 全体のスナップショット
 - 差分を受け取るために、スナップショットを取った位置のポジション情報
- => データプレーンのテーブル再構築、追従に使える

- 起動時に、まずDBのポジションを取得
- 必要データをselect
- selectしている間にもポジションが進む
- selectしたものを自分のデータの中に書き込む
- 確保しておいたポジションから後ろを自分のデータに適用し続ける
 - 2度漬けしても問題にないのでこれが許される

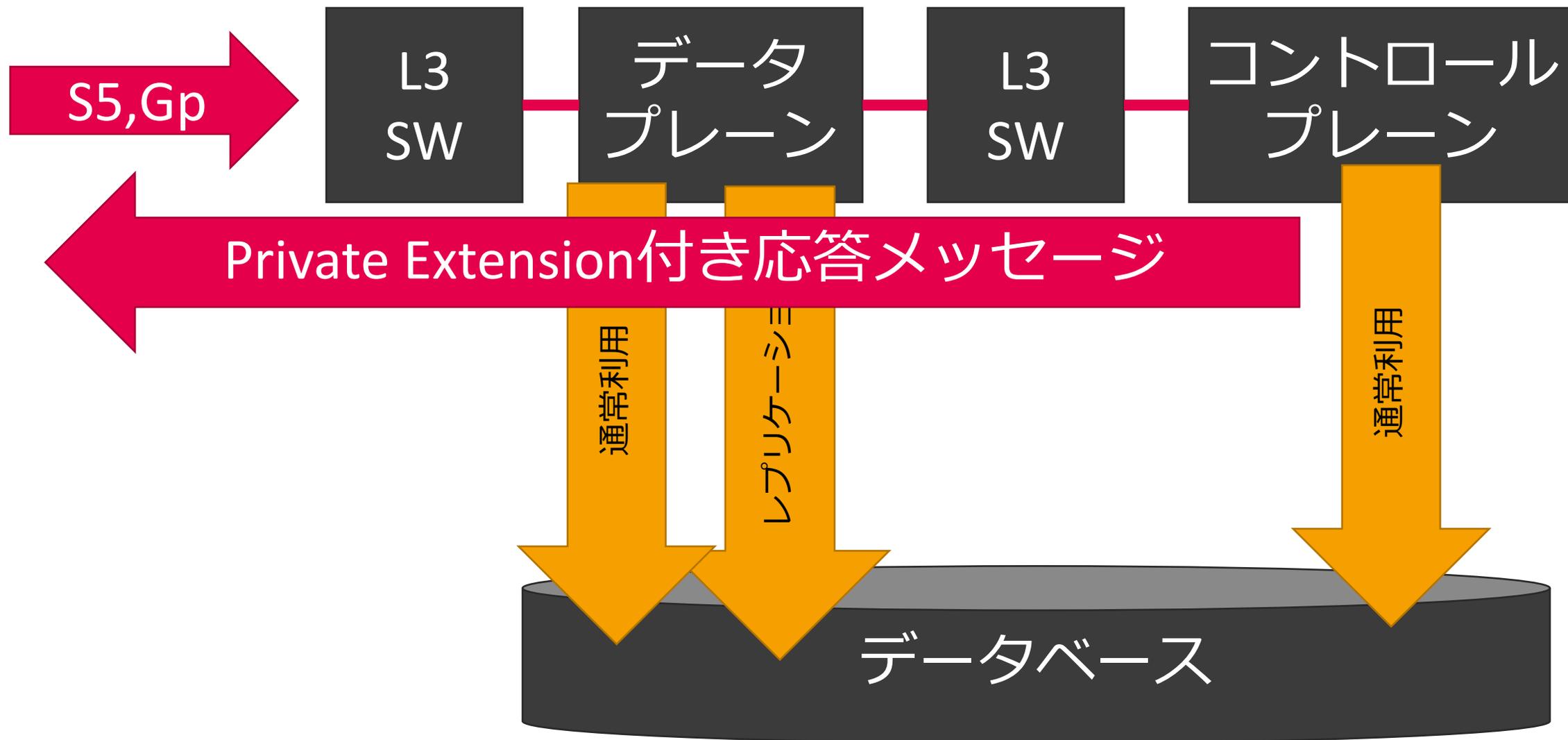


- GTPv1、GTPv2、GTP-U(echo,error,ICMPv6 RS)の実装
- キャリア様で公開されている技術条件集に準拠
- IPv4,IPv6,IPv4v6で動作
- PCOでのDNSは8.8.8.8などを設定
- IPv6を付与した時のICMPv6 Router Solicitationに応答する
- Private Extensionを付与してレスポンスを作る

Private Extension IEの実装したもの



- 実際使う時はvender idをianaサイトで登録して使ってください。snmp でも使えます！
- OSSの中では0で書いてあります。



- レスポンスに含めたかった
- リクエストにしかない情報があった
 - SGWのアドレスなどなど
- まとめてレスポンスに詰めれば必要な情報が揃う
- プロトコル仕様上問題ないところに情報を入れたかった

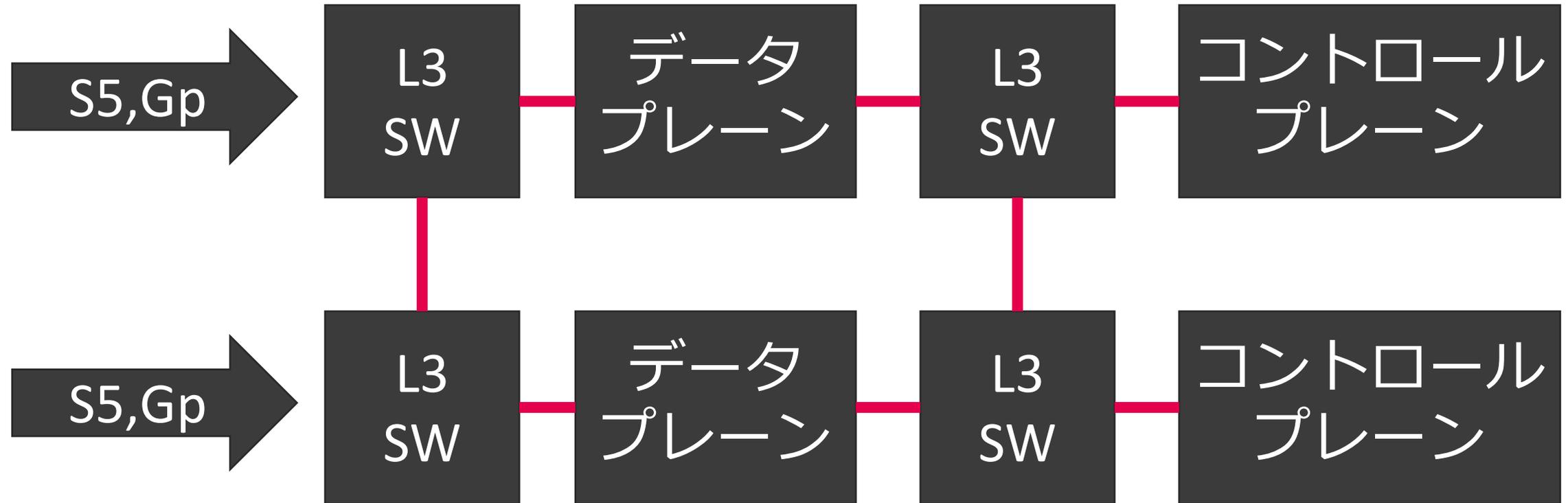
- DBレプリケーションは遅延がでる
- データプレーン反映前にGTP-Uがくるとまずい
- GTP-Cのレスポンスに混ぜて即時反映すればいい

- DPDKには物理ポート2portをアサイン
- 基本的には2ポートをブリッジするスルー動作をする
- このブリッジはパケットを破棄したり、書き換えたりする
 - 興味のある処理だけに限定して書く GTP encap/decapなど



- データベースでC/D分離されているからテストしやすい
 - 各プレーンで独立してテスト可能
- コントロールプレーンのCI試験
 - リクエストに対するDB書き変わり
 - 応答チェック。Private Extension含む
- データプレーンのCI試験
 - DB書き代わりに応じた内部テーブル更新
 - PrivateExtensionを見た内部テーブル更新
 - 内部テーブルを使ったGTPトンネルヘッダの付け外し

- gtestを使ったテストを記載
 - そのためにC++
- DPDKでもわかりやすいテストが描けるように工夫
 - 入れるパケットを定義して、ロジックを通し、出力をチェックする
 - DPDKで動くハードウェアがない状態でもテストできるようにした



右から左にデータプレーンを超えるところと
左から右にデータプレーンを超えるところを一致させたい
BGPならLP等で強制する

- 昔作ったPGWをちょっと整理して公開しました
- だいぶ試行錯誤したデザイン思想です
- 昔はもっといろいろなものがついてました
 - 帯域制御周りのあれこれ
 - 公平制御のためのあれこれですv6頑張った
 - v4とv6が混ざると公平制御が辛かった
 - データプレーンがマイクロサービスかされてチェーンされてました

- 5GCのUPF実装
- できるならRU/DU/CU以外は自前にしたい
- ローカル5G金かかる