次世代スマートシティに向けた 5G・IoT プロジェクトの取り組み

Cooperation with 5G and IoT Projects in Okinawa Open Laboratory
Towards Next-Generation Smart City

OOL IoT プラットフォーム PJ
阿部 宝史
一般社団法人沖縄オープンラボラトリ
NECソリューションイノベータ株式会社

OOL 5G PJ
角田 佳史
一般社団法人沖縄オープンラボラトリ
NTT Communications 株式会社

2020.12.10　12:00 - 12:45

アジェンダ

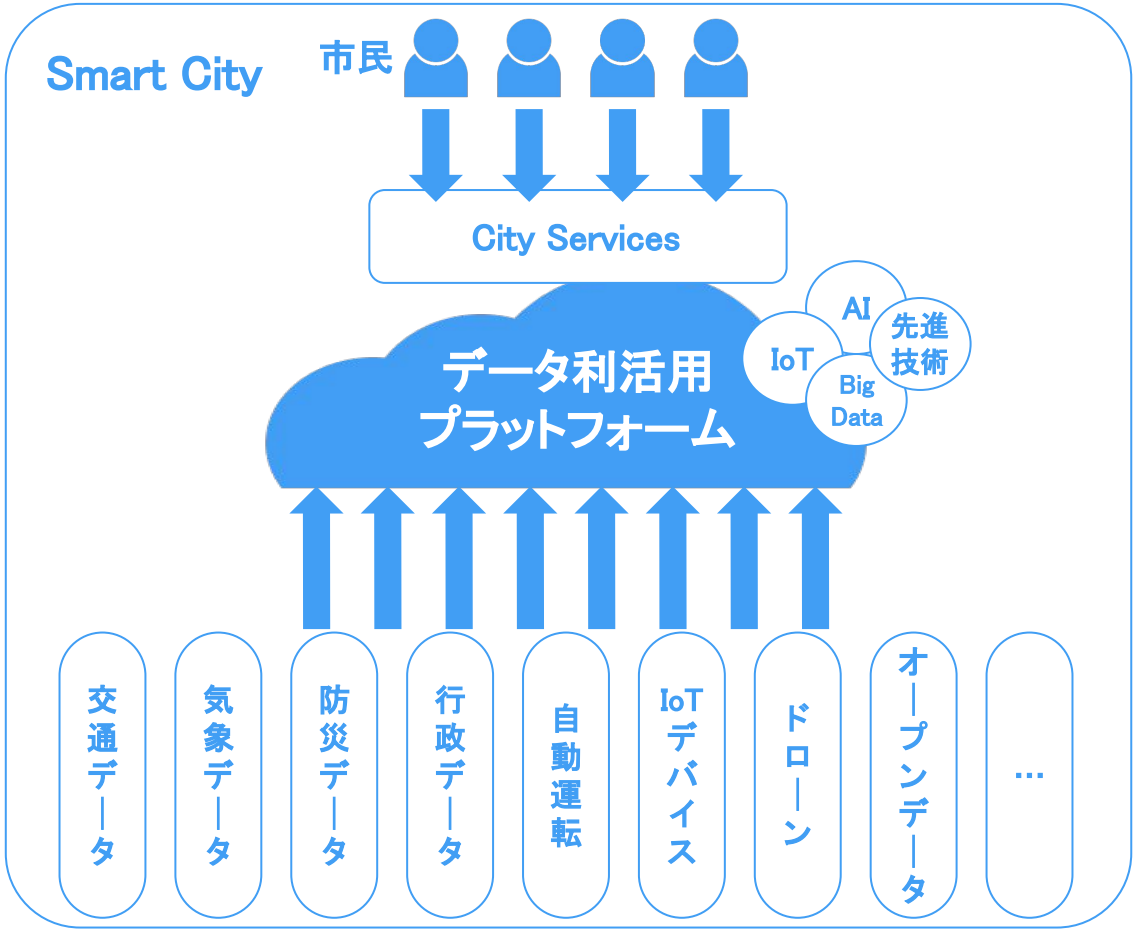
- はじめに
  - スマートシティとは
  - スマートシティの課題
  - 課題に対してのアプローチ
- IoTプロジェクト関連
  - FIWAREとは
  - Meteoroidの開発
  - Meteoroidユースケース検証
  - ユースケースデモ
- 5Gプロジェクト関連
  - 5GPJのアプローチ
  - Free5GCについて
  - 検証環境について
  - 簡易デモ

スマートシティとは

市民生活や行政などの
まちづくりに先端技術を適用

都市全体を効率化

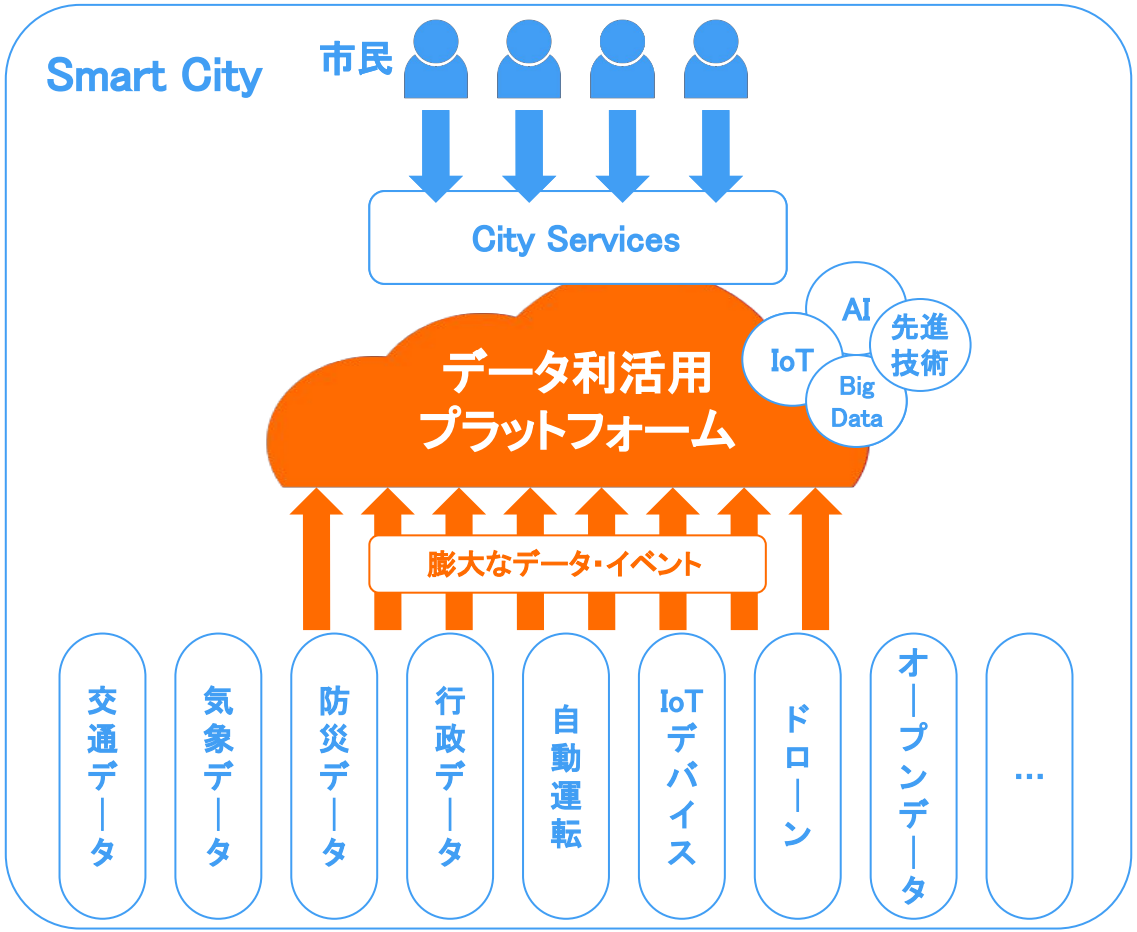
様々な業種間における
データ流通



スマートシティにおける様々な技術課題(インフラ視点)

1. 多種多様なデータを管理し、サービス開発を行う事ができるプラットフォームは？

2. 大量のデータやリアルタイム性のあるイベントを処理するためのネットワークは？



課題への解決アプローチ

1. 多種多様なデータを管理し、サービス開発を行う事ができるプラットフォームは？
    - サービス開発を容易にできるように、データ利活用プラットフォームFIWAREとFunction as a Serviceとの連携を検証
2. 大量のデータやリアルタイム性のあるイベントを処理するためのネットワークは？
    - 5G Multi Access Edge Computing (MEC) を活用
        - 物理的にデバイスや基地局に近い場所にコンピューティングリソースを配置
        - クラウドまで上げる必要のないデータやイベントは MEC で処理

IoT プラットフォーム PJ 編

~FIWARE とFaaS の連携したユースケース紹介

IoT プラットフォーム PJ のアプローチ

IoTプラットフォーム PJ におけるアプローチ

- FIWAREとFunction as a Service (FaaS)を連携したMeteoroidの開発
  - FaaS (Apache OpenWhisk) を利用して、
    FIWARE エコシステムにおけるイベントドリブンアプリケーション実行環境を提供

- Meteoroidを利用したユースケース検証を 5G PJと連携して実施
  - ユースケース: 人物属性情報を基にした広告サイネージ制御

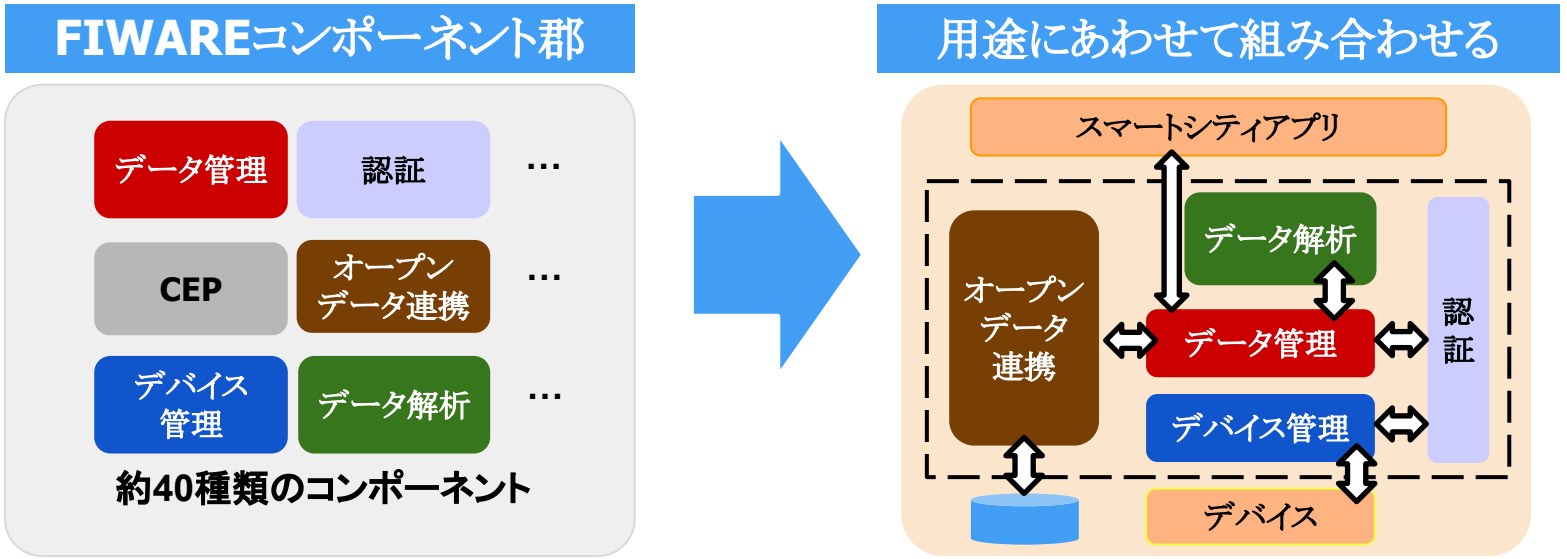
FIWARE とは

- IoT アプリケーション開発のためのソフトウェアの集合体
- 次世代インターネット官民連携プログラム@EU で開発
- 次世代インターネット技術における EU の競争力強化と社会/公共分野のスマートアプリケーション開発を支援
- 欧州の自治体や、スタートアップで利用（150都市、31カ国で利用）
  - スペイン/サンタンデール市　スマートごみ収集
  - 日本/高松市　河川水位測定(防災)
  - etc...

FIWARE
Open APIs for Open Minds

FIWARE とは

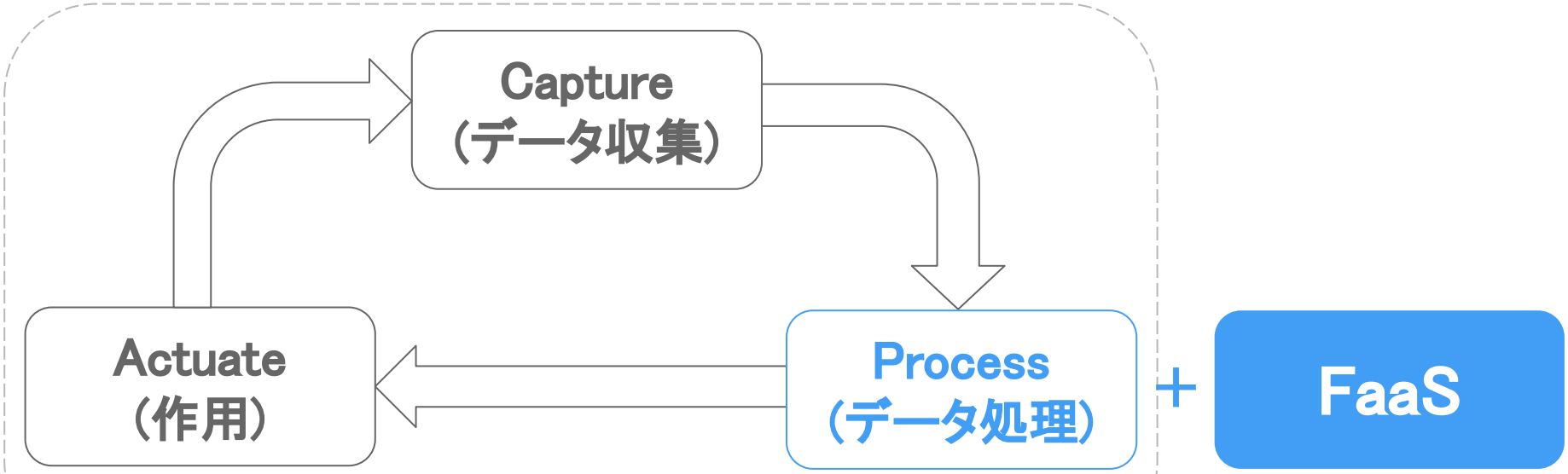
- オープンソースソフトウェア (OSS)
- 各コンポーネントは標準化された API (NGSI) に準拠
- 必要なコンポーネントを自由に組み合わせて利用



FIWARE のデータ活用サイクルと課題

FIWARE のデータ活用サイクルで、Process (データ処理) の機能が不足
Function as a Service (FaaS) を組み合わせることで、Processの機能を強化

※FaaSとは、イベントドリブン型のサーバーレスアプリケーション実行環境

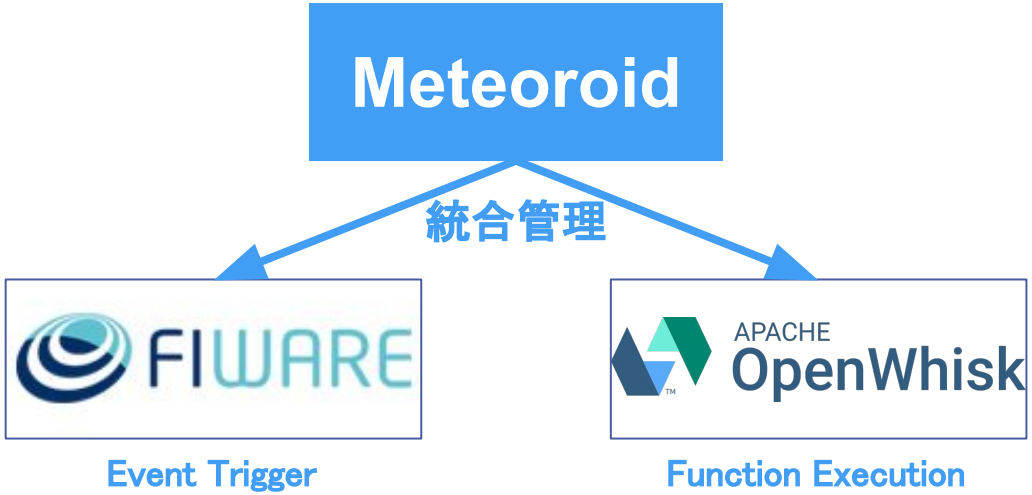


FIWAREにおけるデータ活用サイクル

参考: https://www.fiware.org/developers/

Meteoroid とは

Meteoroid とは、FIWARE (IoTプラットフォーム)と Apache OpenWhisk (FaaSプラットフォーム)を統合管理し、FIWARE エコシステム内におけるイベントドリブン処理を実現する OOL 発の OSS



Apache OpenWhisk とは…

2016年にIBM が公開した OSS の Function as a Service プラットフォーム

Meteoroid ユースケース検証の必要性

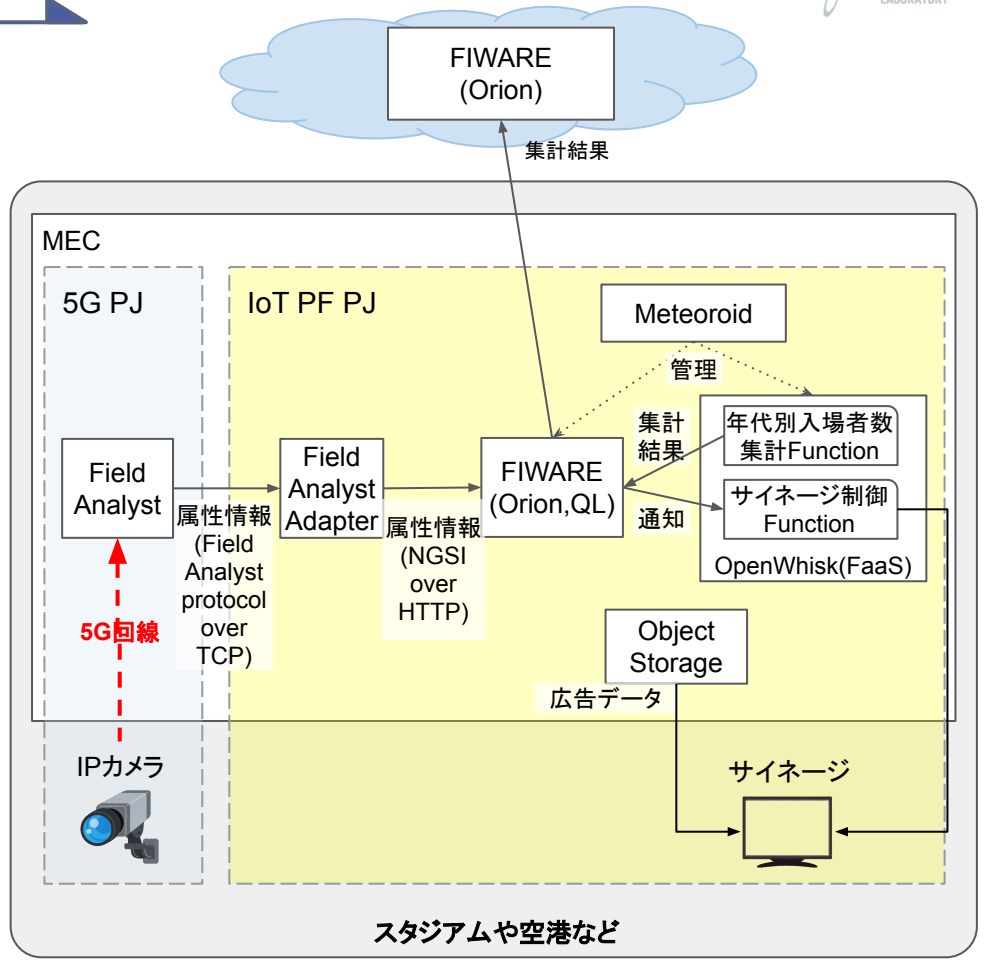
実際のユースケースに Meteoroid を利用することで、課題が解決できるか検証する必要がある

- FIWARE エコシステムでイベントドリブン処理が実現できるか
- データ利活用プラットフォーム FIWARE におけるアプリケーション開発が容易になるかどうか

IoT × 5G ユースケースイメージ

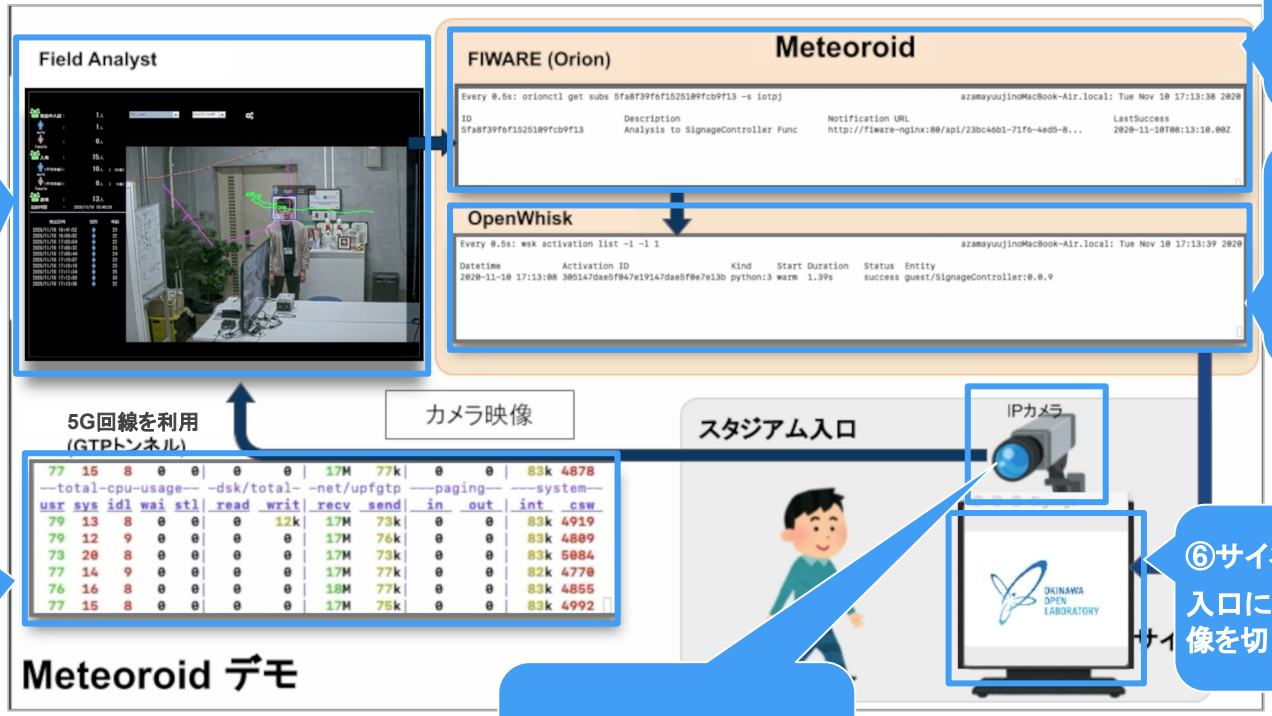
ユースケース

- スタジアムや空港の入り口を想定
- 入場者の属性(年齢)に応じた広告をサイネージ表示
- 個々人の属性情報をField Analyst で収集し、MEC の FIWARE へ3秒間隔で通知・蓄積。
- 属性情報に従いFunction でサイネージを制御。また、年代別入場者数を1日1回集計。
- 集計結果はクラウドのFIWARE に保存。広告データ(画像)は Object Storage から取得。



Meteoroid Demo

Meteoroid を使ってサイネージを制御するデモ



③Field Analyst

カメラ映像から人物の年齢や性別などの属性情報を検出

② 5G 回線の通信状態

④ FIWARE Orion

データブローカーとして、データの収集や蓄積、外部システムへのイベント通知を担う

⑤ OpenWhisk(FaaS)

FaaS として FIWAREから受け取った属性情報を基にサイネージを制御

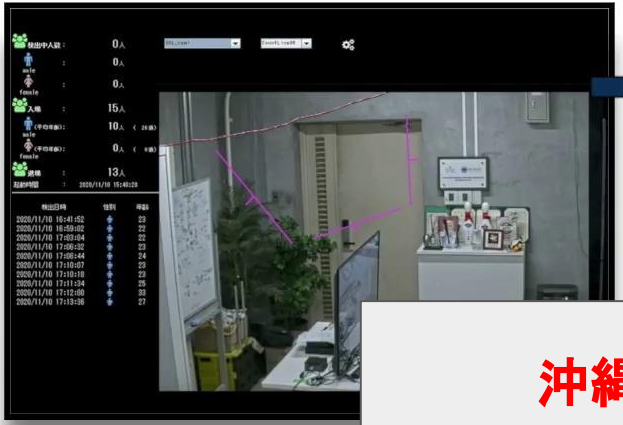
⑥サイネージ

入口に設置されており、表示画像を切り替える API を持つ

①ネットワークカメラ

入口で人物を撮影

Field Analyst



Meteoroid

FIWARE (Orion)

```
Every 0.5s: orionctl get subs 5fa8f39f6f1525109fcb9f13 -s iotpj                  azamayuujinoMacBook-Air.local: Tue Nov 10 17:14:03 2020

ID                        Description                    Notification URL                                  LastSuccess
5fa8f39f6f1525109fcb9f13  Analysis to SignageController Func  http://fiware-nginx:80/api/23bc46b1-71f6-4ed5-8...  2020-11-10T08:13:41.00Z
```

OpenWhisk

```
Every 0.5s: wsk activation list -i -l 1                                          azamayuuji

Datetime            Activation ID                  Kind      Start Duration   Status  Entity
2020-11-10 17:13:40 dc0931a06dc445788931a06dc40578c3 python:3 warm  400ms      success guest/SignageControll
```

OpenWhisk からFIWAREへ 通知完了時間

沖縄オープンラボラトリのブース動画で
閲覧することができます

5G回線を利用
(GTPトンネル)

```
 59  19  22   0   0|   0     0 |  13M   70k|   0     0 |  70k 4358
--total-cpu-usage-- -dsk/total- -net/upfgtp ---paging-- ---system--
usr sys idl wai stl| read  writ| recv  send|  in   out | int   csw
 62  16  21   0   0|   0     0 |  14M   79k|   0     0 |  72k 4461
 76  19   5   0   0|   0     0 |  18M   77k|   0     0 |  79k 3607
 82  15   3   0   0|   0   40k |  19M   67k|   0     0 |  83k 4137
 72  19   9   0   0|   0     0 |  17M   68k|   0     0 |  85k 5572
 74  19   7   0   0|   0     0 |  17M
 75  17   8   0   0|   0     0 |  17M
```

データの流れを時系列で見てみます。
OpenWhiskのファンクション実行開始時間13m08s
にDuration 1.39sを足した時間がOrionの通知完了時
間 13m10s に近いことがわかります。

サイネージ

Meteoroid デモ

ユースケース検証から得られた知見

- **Meteoroid を活用したユースケース検証**
  - Meteoroidによって、アプリケーションを容易に開発/実行
    - FIWAREとFaaSの連携によってインフラ構築を必要とせず、Function開発のみでアプリケーションを実行できた
    - Functionの登録や削除、FIWAREからFunctionのデータ通知設定がMeteoroid CLIのみで行うことができた

    → スマートシティに向けたサービス開発を行えるプラットフォームとしてMeteoroidは有用性がある

  - ユースケース検証から得られた今後の方針予定
    - Meteoroid CLIでの不足している機能の補完するドキュメントを作成

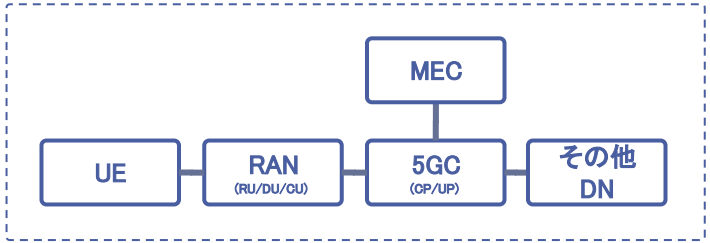
OKINAWA
OPEN
LABORATORY

5Gプロジェクト編

5G PJ のアプローチ

大容量データや低遅延処理に対応するためのネットワークは？

- **5Gにおける特徴など**
  - Multi Access Edge Computing (MEC)
    - デバイス近傍にあるエッジ上で様々な処理が可能
  - Network Slicling
    - 用途毎に柔軟なリソース制御が可能
- **5G検証におけるアプローチ**
  - 5GS (5G System) の用意
    - 5GC: **Free5GC**
    - RAN: 製品版/OSS 5G シミュレータ
    - MEC: **FIWARE**
  - その他の技術領域にも着目
    - オーケストレータ・コントローラ、MECプラットフォーム・アプリなど
  - IoTプロジェクト連携向、5Gプロジェクト向けに複数検証環境を用意



Free5GCについて

- 台湾の国立交通大学(NCTU) が主導で開発
- **3GPP Release 15 準拠・SA (Stand-Alone) 構成対応**
  - 現在 Stage3 v3.0.4 までリリースされており、Stage3からSA構成対応
- 様々な Network Function に対応
  - AMF, SMF, UPF, NRF, NSSF, AUSF, UDR, UDM, PCF
  - **スライシング機能・多段UPF・ULCL (UpLink Classifier)** などの一部実装有り
- 主要なコールシーケンスに対応
  - UE Registration/Deregistration, PDU Session Establishment など
  - 実装上のバグなどで対応していないメッセージや機能など未だ多く有り
- コミュニティの動きも比較的活発でGithub での注目度も高い
  - Star の数は 700 以上
- その他のOSS 5GCoreは以下
  - open5gs, oai-cn5g...

Official: https://www.free5gc.org
Github: https://github.com/free5gc/free5gc

OSS 5GCを用いた検証環境 (IoT PJ 向け)

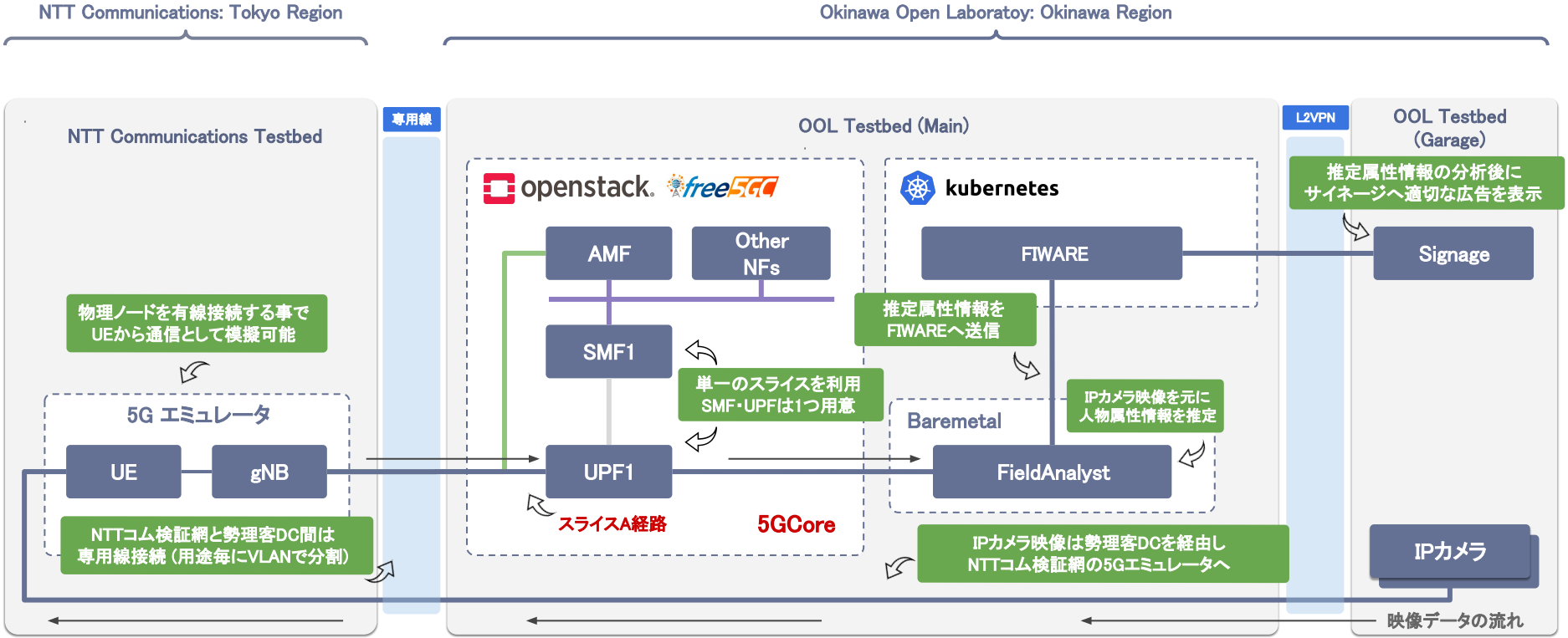
- **IoT PJ 連携に向けた検証環境**
  - 内容
    - 5G網を通じた映像伝送からMEC基盤での映像分析までの一連の流れを再現
  - 検証概要
    - Free5GCore (*1) の利用
    - 製品版5Gエミュレータの利用
    - 上記を通じたFieldAnalyst (*2) (MEC想定) へのIPカメラ映像配信
    - FieldAnalystから属性データの Meteoroid Platform への配信と属性情報分析
      - 分析結果を元にデジタルサイネージへ最適な画像を表示

*1 Free5GCore は OSS 5GCore の一つ

*2 Field Analyst はIPカメラ映像から人物の属性情報の推定を行う

OSS 5GCを用いた検証環境 (IoT PJ 向け)

OpenStack上のFree5GCoreを通じて、Field Analyst へ IPカメラ映像配信を行う検証構成
IoT PJ の FIWARE などと連携し、人物属性情報を元にサイネージへ適切な広告を表示

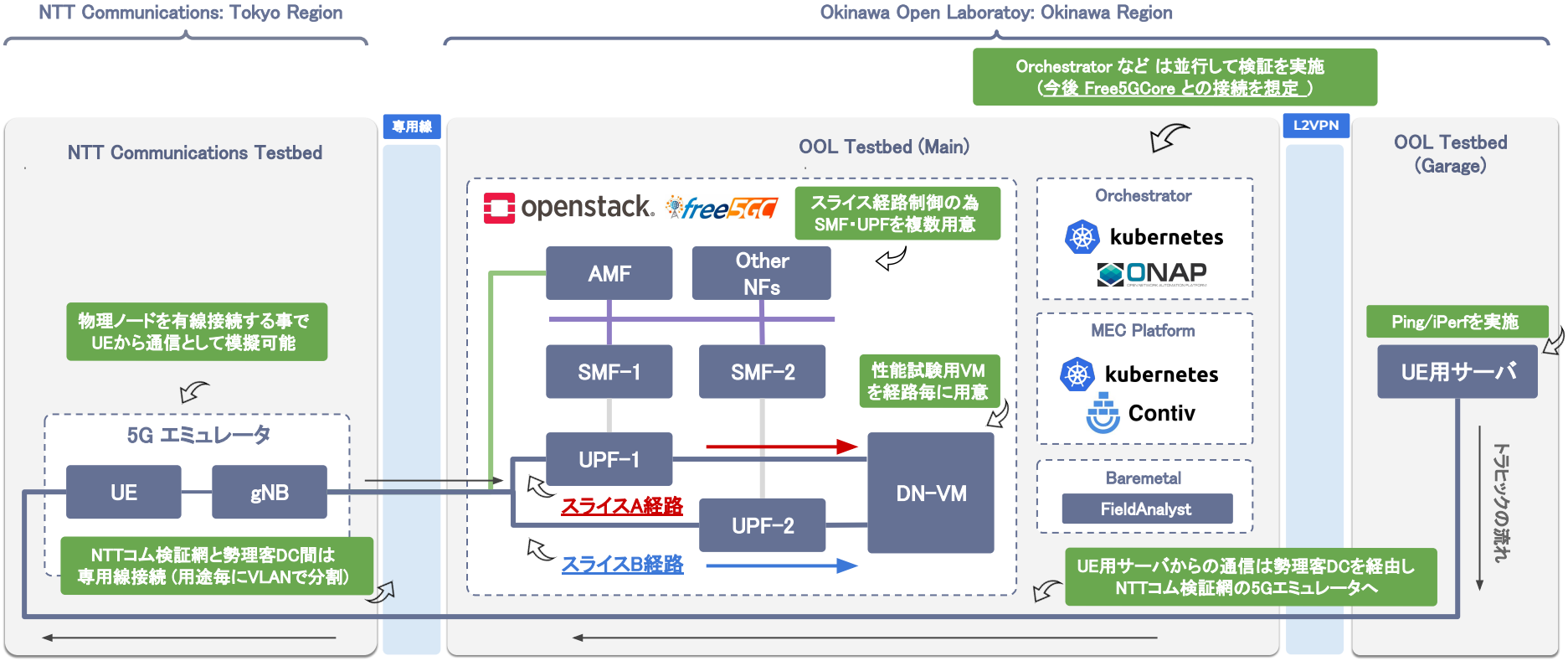


OSS 5GCを用いた検証環境 (5G PJ 向け)

- **5G PJ 単体検証に向けた検証環境**
  - 内容
    - 5G網におけるスライスを用いた経路制御及び性能測定評価を実施
  - 検証概要
    - Free5GCoreの利用
    - 製品版5Gエミュレータの利用
    - 複数スライスを用いた経路制御
      - 経路毎にTCコマンド (Traffic Control) を用いてQoS制御を擬似的に再現
    - スループットなど各種性能試験の実施
      - iPerf の利用、TCPで測定、ブロックサイズはデフォルト値

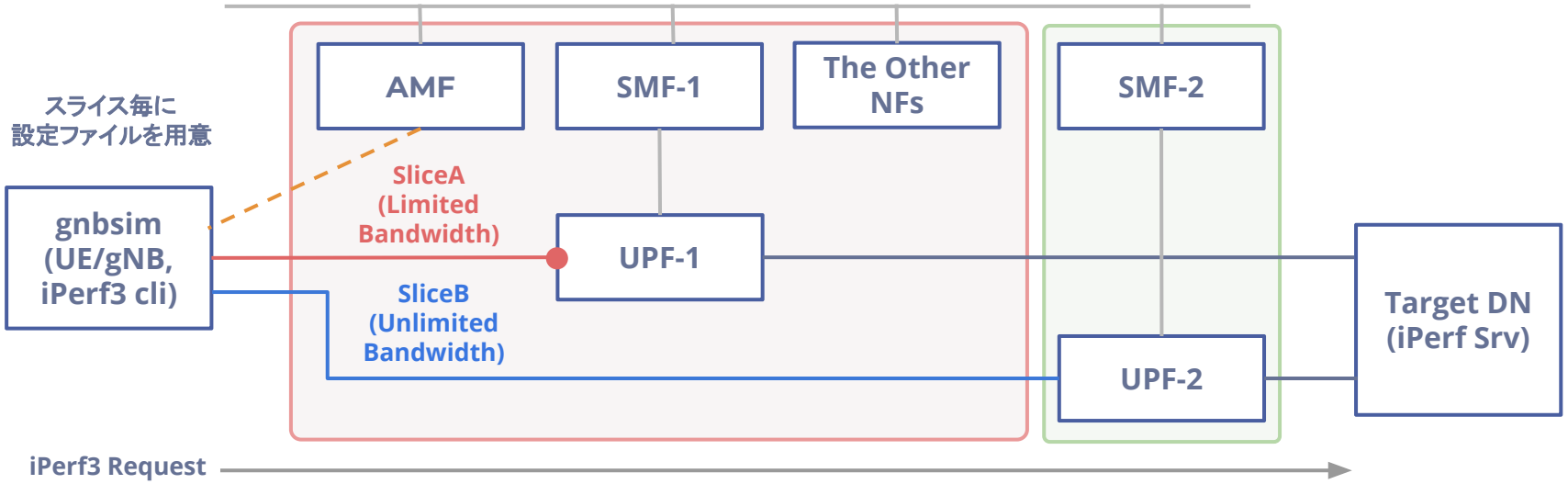
OSS 5GCを用いた検証環境 (5G PJ 向け)

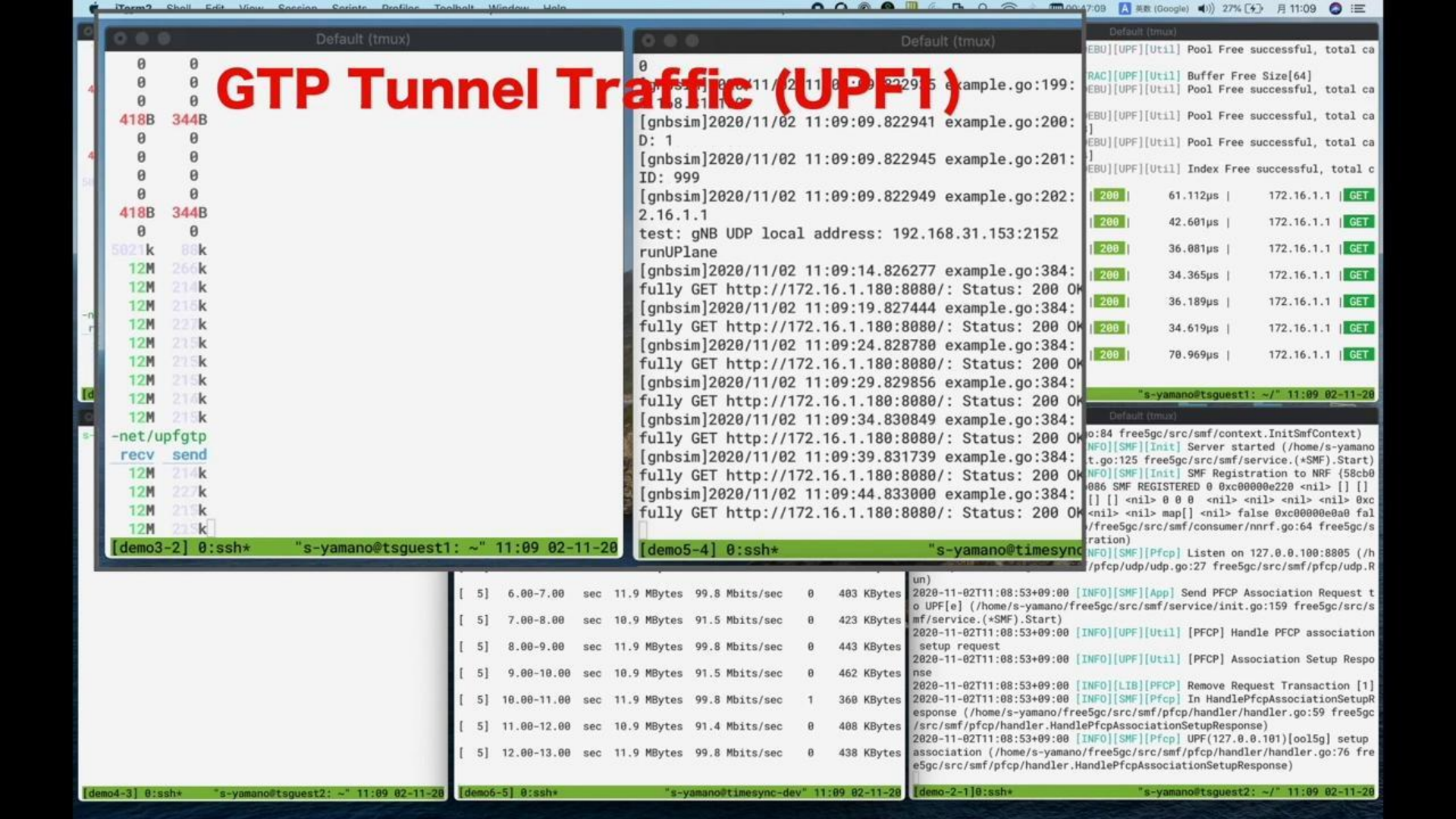
OpenStack上のFree5GCoreを用いて複数スライスを用いた経路制御を行う検証構成
UE用サーバから5Gエミュレータ・Free5GCoreを通じて、各種性能試験を実施



OSS 5GCを用いたスライスによる経路制御デモ

- UE/gNB には OSS 5G シミュレータの ***gnbsim(OSS)*** を利用
- 5GCore には ***Free5GC v3.0.4*** を利用
- スライス毎の経路選択を行い、TCコマンドにより擬似的にQoSを適用
  - スライスによる経路制御（SMF・UPF選択）のために修正パッチを作成・適用




GTP Tunnel Traffic (UPF1)
Default (tmux)
0 0
0 0
0 0
418B 344B
0 0
0 0
0 0
0 0
418B 344B
0 0
-net/upfgtp
recv send
[demo3-2] 0:ssh* "s-yamano@tsguest1: ~" 11:09 02-11-20
Default (tmux)
0
[gnbsim]2020/11/02 11:09:09.822941 example.go:200:
D: 1
[gnbsim]2020/11/02 11:09:09.822945 example.go:201:
ID: 999
[gnbsim]2020/11/02 11:09:09.822949 example.go:202:
2.16.1.1
test: gNB UDP local address: 192.168.31.153:2152
runUPlane
[gnbsim]2020/11/02 11:09:14.826277 example.go:384:
fully GET http://172.16.1.180:8080/: Status: 200 OK
[gnbsim]2020/11/02 11:09:19.827444 example.go:384:
fully GET http://172.16.1.180:8080/: Status: 200 OK
[gnbsim]2020/11/02 11:09:24.828780 example.go:384:
fully GET http://172.16.1.180:8080/: Status: 200 OK
[gnbsim]2020/11/02 11:09:29.829856 example.go:384:
fully GET http://172.16.1.180:8080/: Status: 200 OK
[gnbsim]2020/11/02 11:09:34.830849 example.go:384:
fully GET http://172.16.1.180:8080/: Status: 200 OK
[gnbsim]2020/11/02 11:09:39.831739 example.go:384:
fully GET http://172.16.1.180:8080/: Status: 200 OK
[gnbsim]2020/11/02 11:09:44.833000 example.go:384:
fully GET http://172.16.1.180:8080/: Status: 200 OK
[demo5-4] 0:ssh* "s-yamano@timesync
Default (tmux)
EBU][UPF][Util] Pool Free successful, total ca
RAC][UPF][Util] Buffer Free Size[64]
EBU][UPF][Util] Pool Free successful, total ca
EBU][UPF][Util] Pool Free successful, total ca
EBU][UPF][Util] Pool Free successful, total ca
EBU][UPF][Util] Index Free successful, total c
200 61.112µs 172.16.1.1 GET
200 42.601µs 172.16.1.1 GET
200 36.081µs 172.16.1.1 GET
200 34.365µs 172.16.1.1 GET
200 36.189µs 172.16.1.1 GET
200 34.619µs 172.16.1.1 GET
200 70.969µs 172.16.1.1 GET
"s-yamano@tsguest1: ~/" 11:09 02-11-20
Default (tmux)
o:84 free5gc/src/smf/context.InitSmfContext)
NFO][SMF][Init] Server started (/home/s-yamano
t.go:125 free5gc/src/smf/service.(*SMF).Start)
NFO][SMF][Init] SMF Registration to NRF {58cb0
086 SMF REGISTERED 0 0xc00000e220 <nil> [] []
[] [] <nil> 0 0 0 <nil> <nil> <nil> <nil> 0xc
<nil> <nil> map[] <nil> false 0xc00000e0a0 fal
/free5gc/src/smf/consumer/nnrf.go:64 free5gc/s
ration)
NFO][SMF][Pfcp] Listen on 127.0.0.100:8805 (/h
/pfcp/udp/udp.go:27 free5gc/src/smf/pfcp/udp.R
un)
2020-11-02T11:08:53+09:00 [INFO][SMF][App] Send PFCP Association Request t
o UPF[e] (/home/s-yamano/free5gc/src/smf/service/init.go:159 free5gc/src/s
mf/service.(*SMF).Start)
2020-11-02T11:08:53+09:00 [INFO][UPF][Util] [PFCP] Handle PFCP association
setup request
2020-11-02T11:08:53+09:00 [INFO][UPF][Util] [PFCP] Association Setup Respo
nse
2020-11-02T11:08:53+09:00 [INFO][LIB][PFCP] Remove Request Transaction [1]
2020-11-02T11:08:53+09:00 [INFO][SMF][Pfcp] In HandlePfcpAssociationSetupR
esponse (/home/s-yamano/free5gc/src/smf/pfcp/handler/handler.go:59 free5gc
/src/smf/pfcp/handler.HandlePfcpAssociationSetupResponse)
2020-11-02T11:08:53+09:00 [INFO][SMF][Pfcp] UPF(127.0.0.101)[ool5g] setup
association (/home/s-yamano/free5gc/src/smf/pfcp/handler/handler.go:76 fre
e5gc/src/smf/pfcp/handler.HandlePfcpAssociationSetupResponse)
[demo-2-1]0:ssh* "s-yamano@tsguest2: ~/" 11:09 02-11-20
[5] 6.00-7.00 sec 11.9 MBytes 99.8 Mbits/sec 0 403 KBytes
[5] 7.00-8.00 sec 10.9 MBytes 91.5 Mbits/sec 0 423 KBytes
[5] 8.00-9.00 sec 11.9 MBytes 99.8 Mbits/sec 0 443 KBytes
[5] 9.00-10.00 sec 10.9 MBytes 91.5 Mbits/sec 0 462 KBytes
[5] 10.00-11.00 sec 11.9 MBytes 99.8 Mbits/sec 1 360 KBytes
[5] 11.00-12.00 sec 10.9 MBytes 91.4 Mbits/sec 0 408 KBytes
[5] 12.00-13.00 sec 11.9 MBytes 99.8 Mbits/sec 0 438 KBytes
[demo6-5] 0:ssh* "s-yamano@timesync-dev" 11:09 02-11-20
[demo4-3] 0:ssh* "s-yamano@tsguest2: ~" 11:09 02-11-20

5G PJ まとめ

- **オープンな技術を用いたモバイル網の実現に向け**
  技術領域毎に複数プロジェクトを展開し、相互に検証・連携を実施
  - モバイル網構築に際して考慮すべき点は多いので、複数プロジェクトで並行検証

- **3GPP標準仕様に則ったOSSが様々に出てきており、**
  モバイル網を構築する上で一つの選択肢になる可能性がある
  - UE登録・認証・アタッチからワンコールなど必要最小限の機能は整い始めて来ている
    - QoS機能・スライス経路制御など未実装機能やバグは未だ多い
  - 実環境で利用するには関連コードの修正の必要など対応すべき箇所が多い
    - OOLでの検証におけるスライス経路制御（SMF・UPF選択）なども一時パッチを適用

全体まとめ

- **スマートシティの実現に向けて考慮すべき課題が存在**
  - 課題解決に向けたアプローチとしての5G や IoT プロジェクトの取り組み

- **スマートシティにおけるサービス開発を促進する環境は何か**
  - データ利活用プラットフォームFIWAREとFunction as a Service との連携
    - **FIWARE + Function as a Service の連携に向けた Meteoroid** の開発

- **大容量データ・低遅延処理が必要なサービスのためのネットワークは何か**
  - 5Gネットワークの適用
    - **Multi Access Edge Computing (MEC)**
      - デバイス近傍(エッジ)にリソースを配置・処理することにより、低遅延を実現
    - **ネットワークスライシング**
      - 用途ごとにスライスを分離する事により、柔軟にリソースを制御
    - etc...