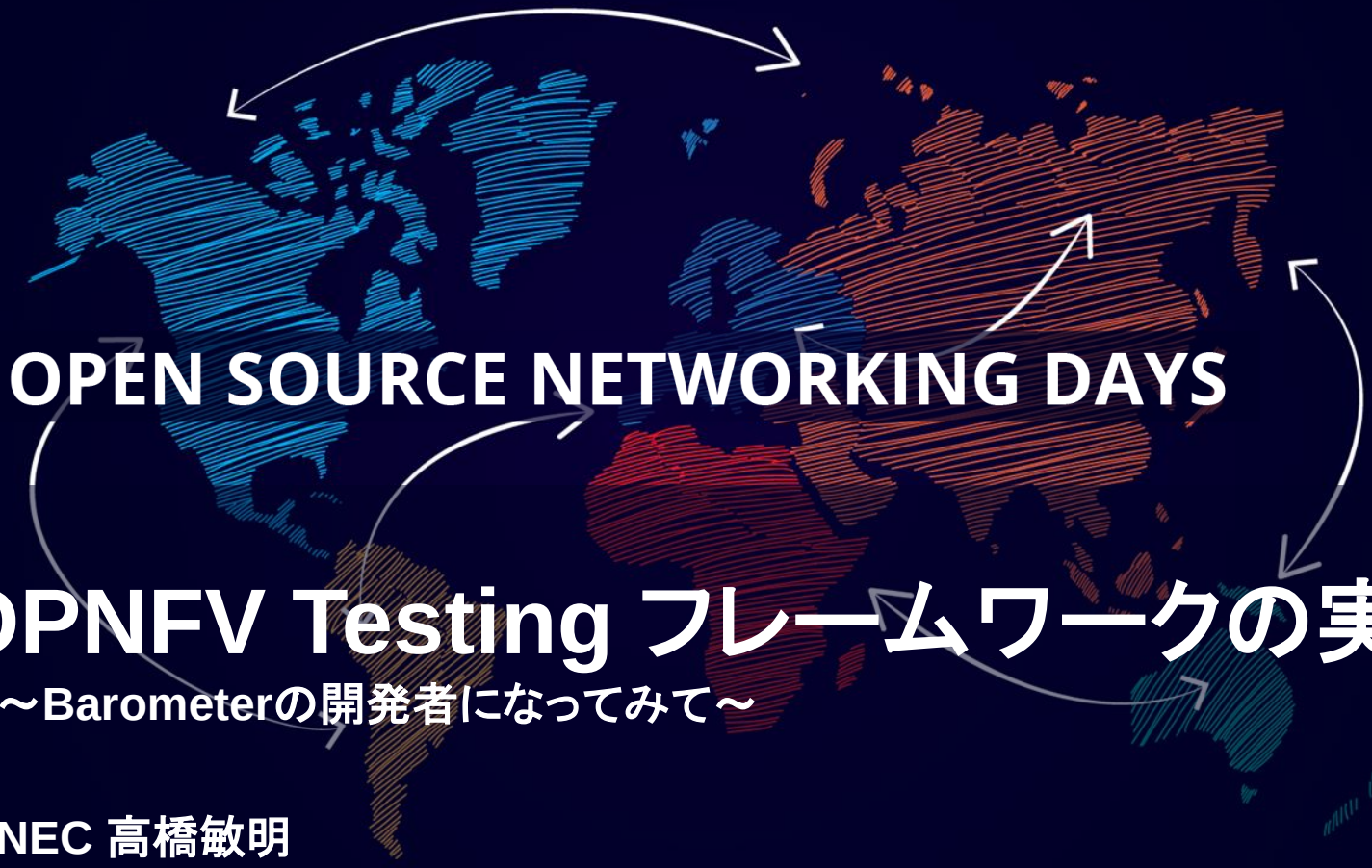


OPNFV Testing フレームワークの実体

～Barometerの開発者になってみて～

NEC 高橋敏明

自己紹介

高橋敏明

- OPNFVとの関わり
  - Doctor デモの開発
  - Barometer Contributor
    - DMAプロジェクトの成果
    - BugFix

※Doctor：OPNFV 障害管理機能プロジェクト
※Barometer：OPNFV 監視機能プロジェクト
※DMAプロジェクト：KDDI総合研究所、Red Hat、NECが進める
分散監視・分析プロジェクト

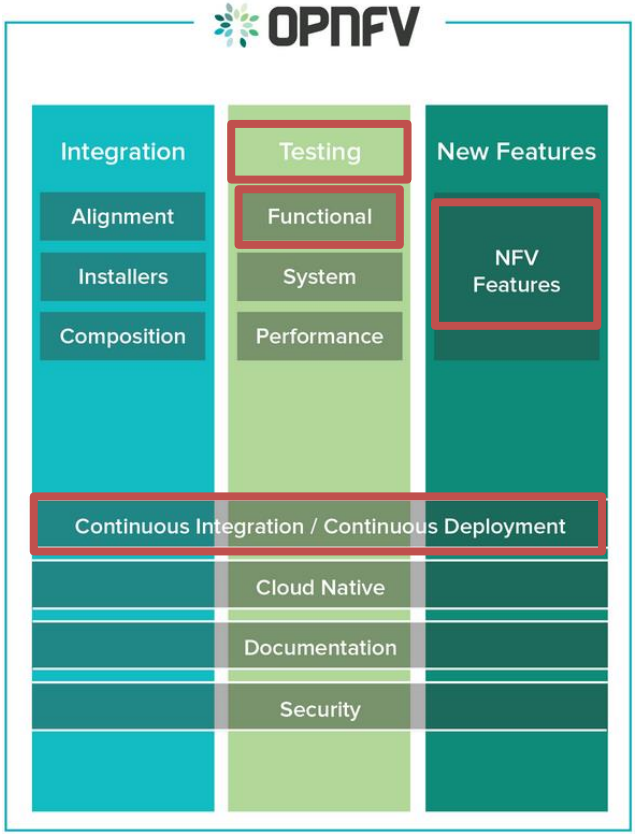
本日のプレゼン内容

Barometer Developerとして
Feature追加をしました。

合わせて、機能のTestingに関わる
部分も開発しました。

これらの経験を通して、
OPNFVのTestingの実状について
お話したいと思います。

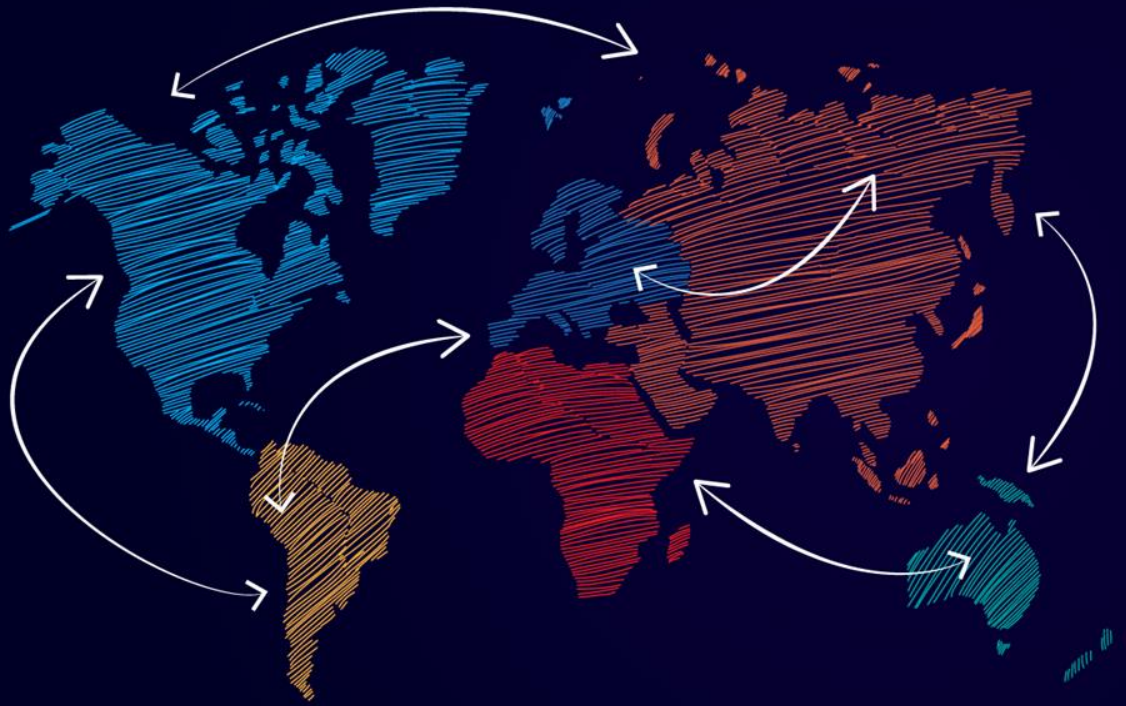
本日の主な話題



アジェンダ

- Testing概要紹介
- 自分たちがやったこと
- トラブルエピソード

- まとめ

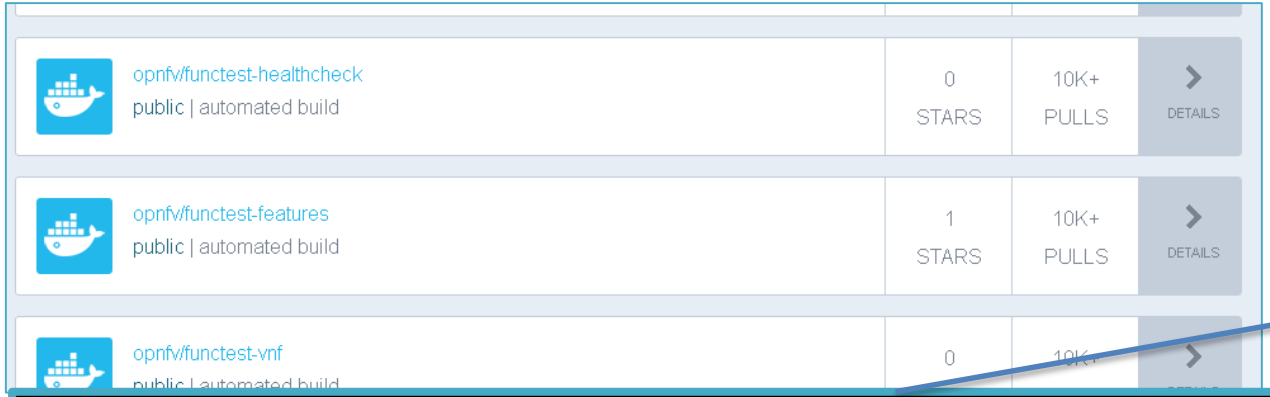


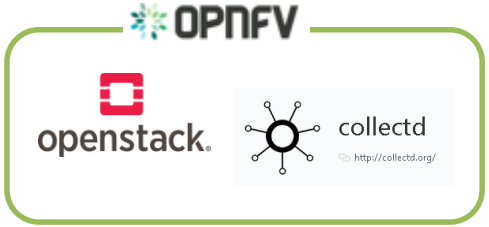
Testing
概要紹介

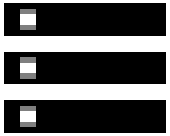
OPNFV Testingが提供するもの

- 機能試験Dockerイメージ
  - Docker run 一発で試験する
- Continuous Integration
  - JenkinsのジョブでCI
- Testing Result Reporting
  - 現状のOPNFV成果物の品質閲覧

機能試験Dockerイメージ





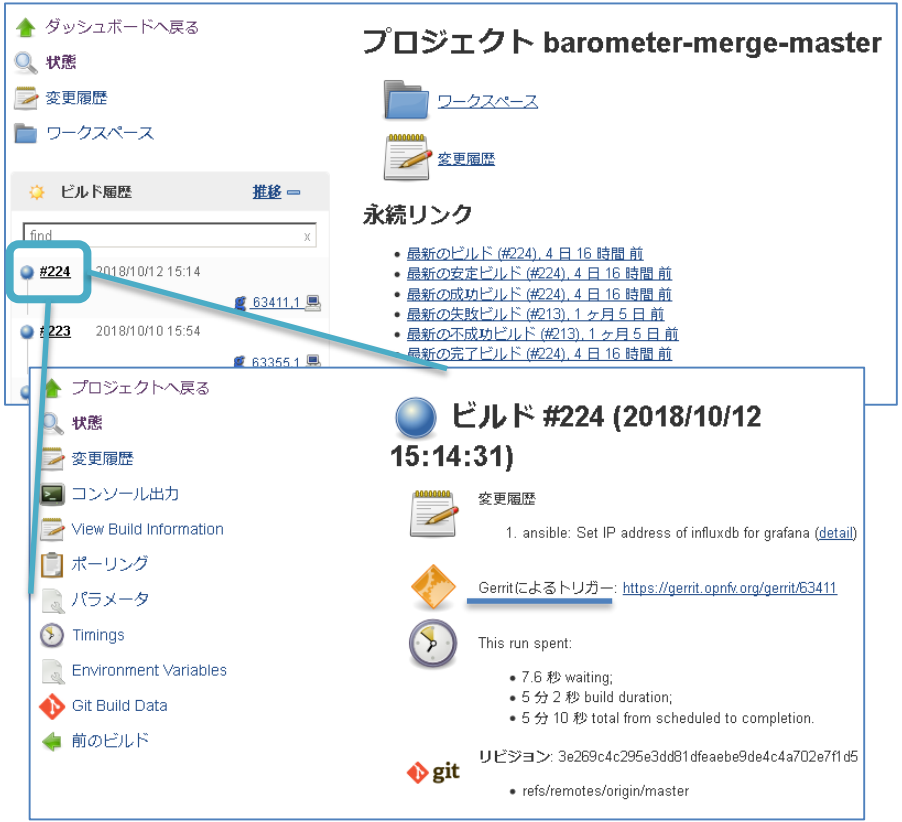


```
# docker run [...] opnfv/functest-features:gambia run_tests -t barometercollectd
......
......
+---------------------------+-------------------+-----------------+----------------+
|         TEST CASE         |      PROJECT      |     DURATION    |     RESULT     |
+---------------------------+-------------------+-----------------+----------------+
|     barometercollectd     |     barometer     |      03:29      |      PASS      |
+---------------------------+-------------------+-----------------+----------------+

2018-10-17 08:32:33,329 - xtesting.ci.run_tests - INFO - Execution exit value: Result.EX_OK
```

Continuous Integration

https://build.opnfv.org/ci/



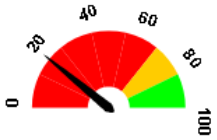
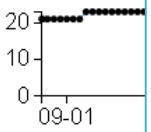
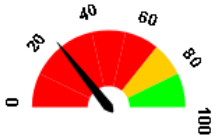
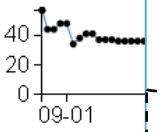
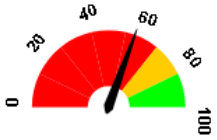
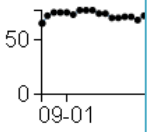
Testing Result Reporting

http://testresults.opnfv.org/reporting/

各デプロイシナリオの
テストOK状況サマリ

シナリオごとの
テストOK状況

List of last scenarios (master) run over the last 10 days

HA Scenario	Status	Trend	Score	Iteration
os-nosdn-ovs_dpdk-ha				
os-odl_l3-nofeature-ha				
os-nosdn-bar-ha				

os-nosdn-bar-ha

healthcheck

connectivity	tenant network 1	tenant network 2	vm ready 1	vm ready 2	single vm 1	single vm 2	vPing (ssh)	vPing (userdata)	cinder tests	api	dhcp

（中略）

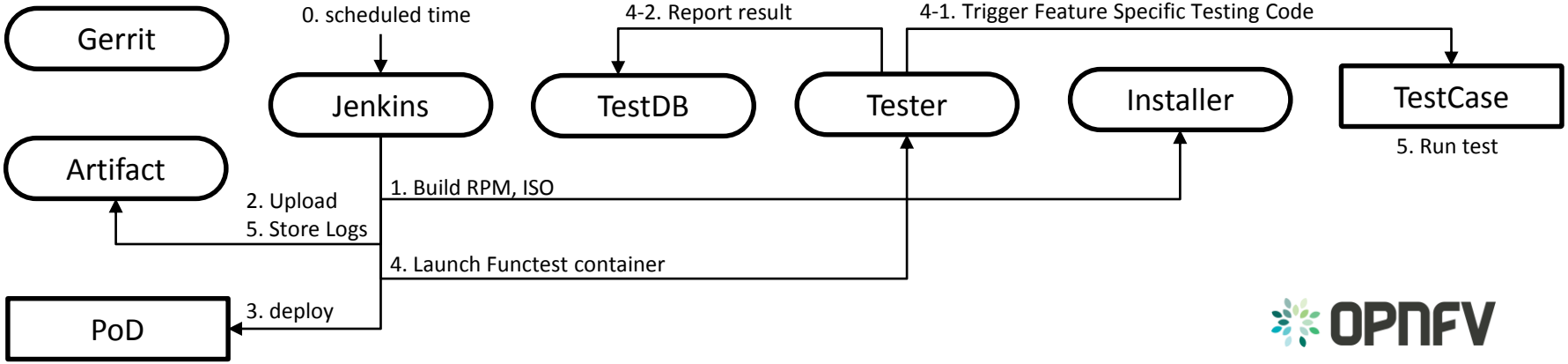
features

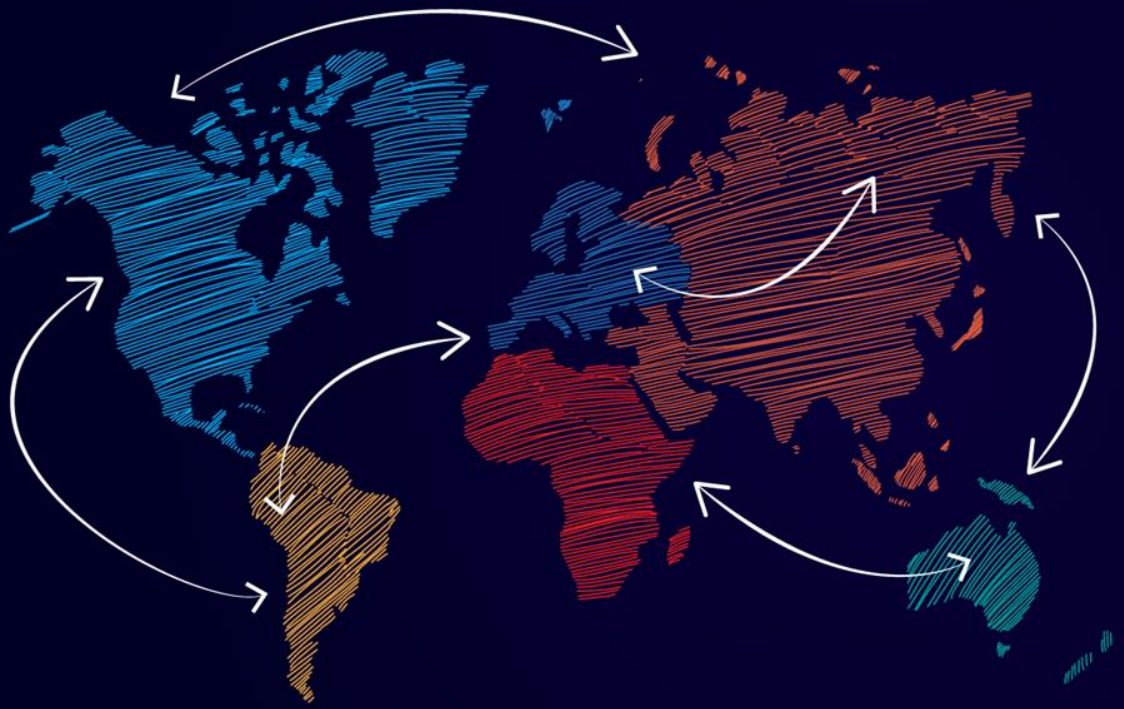
Barometer

参考文献:Doctor壬生さんの資料“OPNFVのコンポーネントと調べ方”
https://www.slideshare.net/r-mibu/opnfv-79129688

例) Apex Daily Job

Pharos / Infra	Releng	Testing Team	Functest	Apex/Fuel	Doctor
putting infra in place	CI Control	Test Coordination	Testing Framework	Integration	Feature Dev.





自分たちが
やったこと
（DMAプロジェクト）

前提

※**青字**が今回追加、変更した部分

- Barometerに対し、1つの機能を追加
  - Barometerリポジトリ
    - src/**dma/*** ←**新規作成、機能コードを配置**

- そのテストの対応が必要
- 今回はOPNFVインストーラチームとは連携しない
  - Optional機能の位置づけ、使う人が手順に従い導入
- Barometerリポジトリ以外に手を入れたくない
  - そのプロジェクトに依頼を投げないといけない
  - マージタイミングがずれる

機能試験Dockerイメージ

※青字が今回追加、変更した部分

- テストコード自体は、Barometerリポジトリにある
  - Barometerリポジトリ
    - src/**dma/***
    - baro_tests**/*.py** ←**DMA用テストを追加、既存一部修正**
    - **requirements.txt** ←**試験にpython-tomlが必要**
      - **※現状は暫定でpython-tomlを使わないコードを入れている**
  - Functestリポジトリ
    - （元々Barometerをbaro_tests配下で試験するよう調整済み）
    - docker/features/Dockerfile ←変更不要

※FunctestのDockerイメージに、python-tomlもちゃんと入っていた

Continuous Integration

※**青字**が今回追加、変更した部分

- Jenkinsが呼ぶコードは、Barometerリポジトリにある
  - Barometerリポジトリ
    - src/**dma/***
    - baro_tests/***.py**
    - **requirements.txt**
    - src/Makefile ←src 配下のものをmake
    - ci/*.sh ←必要なパッケージインストール＋rpmbuild
  - Relengリポジトリ
    - jjb/barometer/barometer.yaml ←元々マージ済み、変更不要
      - Merge時にsrc/Makefile でmake、Dailyでci/*.sh でrpmbuild

漏れです・・・今DMAコードってCIされてないです・・・後でやります

Testing Result Reporting

※**青字**が今回追加、変更した部分

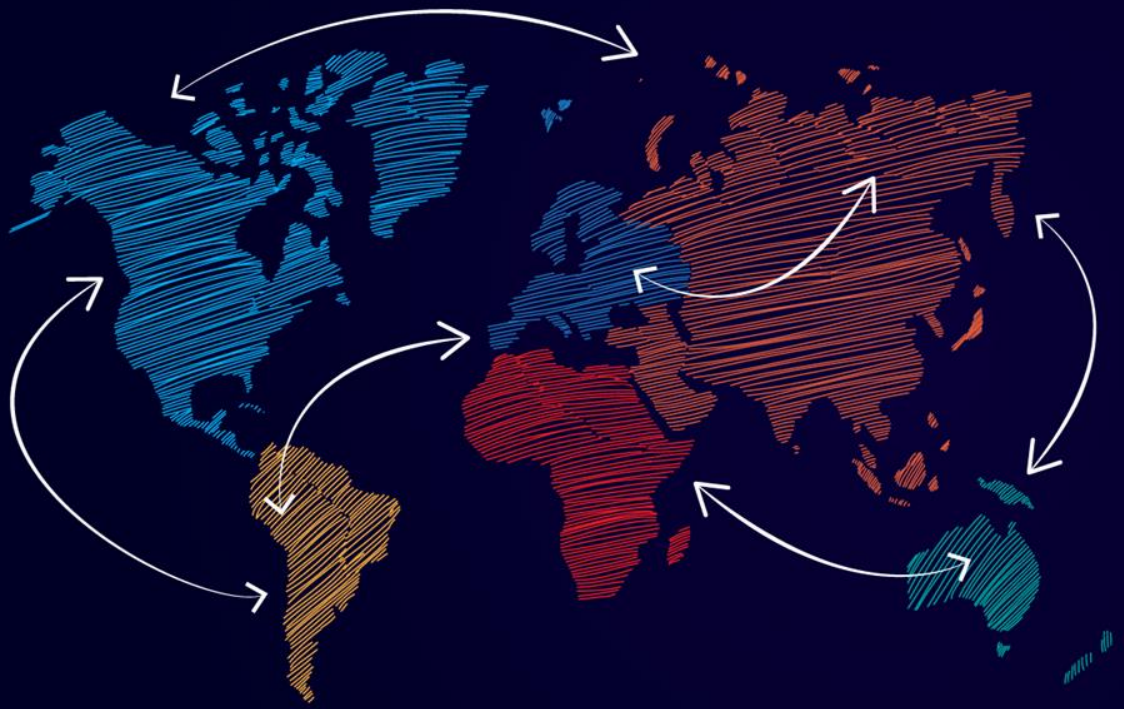
- Barometerでは自分の機能試験コードのみを意識
  - Barometerリポジトリ
    - src/**dma/***
    - baro_tests/***.py** ←**DMA機能が動作している時のみテストするロジック**
    - **requirements.txt**
  - Functestリポジトリ
    - （元々Barometerをbaro_tests配下で試験するよう調整済み）
  - Apexリポジトリ ←今回意識せず
  - Relengリポジトリ
    - （apex-daily：Apexで環境構築してfunctest-dailyを実施）
    - （functest-daily：baro_tests含む機能試験を実施してResultをReport）

ここまでのまとめ

※**青字**が今回追加、変更した部分

- Barometerリポジトリのみの変更でテスト対応完了
  - Barometerリポジトリ
    - src/**dma/***
    - baro_tests/***.py**
    - **requirements.txt**
    - src/Makefile
    - ci/*.sh

✓ 機能試験Dockerイメージ
✓ Continuous Integration（今は漏れ・・・）
✓ Testing Result Reporting（今は未対応）



トラブル
エピソード

ビルドトラブル

我々のコードmerge時に、Jenkins jobでビルドエラー

（ファイル構成一部再掲）

- Barometerリポジトリ
  - src/Makefile　←src 配下のものをmake
  - ci/*.sh　←必要なパッケージインストール＋rpmbuild
- Relengリポジトリ
  - jjb/barometer/barometer.yaml
    - Merge時にsrc/Makefile でmake、Dailyでci/*.sh でrpmbuild

⇒ merge時のmakeは事前にパッケージの依存解決していない
⇒ たまたま我々のmerge時に、依存エラーを出すPodが使われた
※今は使うPodを問題ないものに決め打ち（他メンバが実施）

テストコードのバグ長期内在

Barometerテストコードは一時期、試験が必ずFAILになるバグが内在

（ファイル構成一部再掲）

- Barometerリポジトリ
  - baro_tests/*.py ←ここにバグが内包
  - src/Makefile
  - ci/*.sh
- Relengリポジトリ
  - jjb/barometer/barometer.yaml
    - Merge時にsrc/Makefile でmake、Dailyでci/*.sh でrpmbuild

⇒Barometer内のCI ではFunctestは使われていない

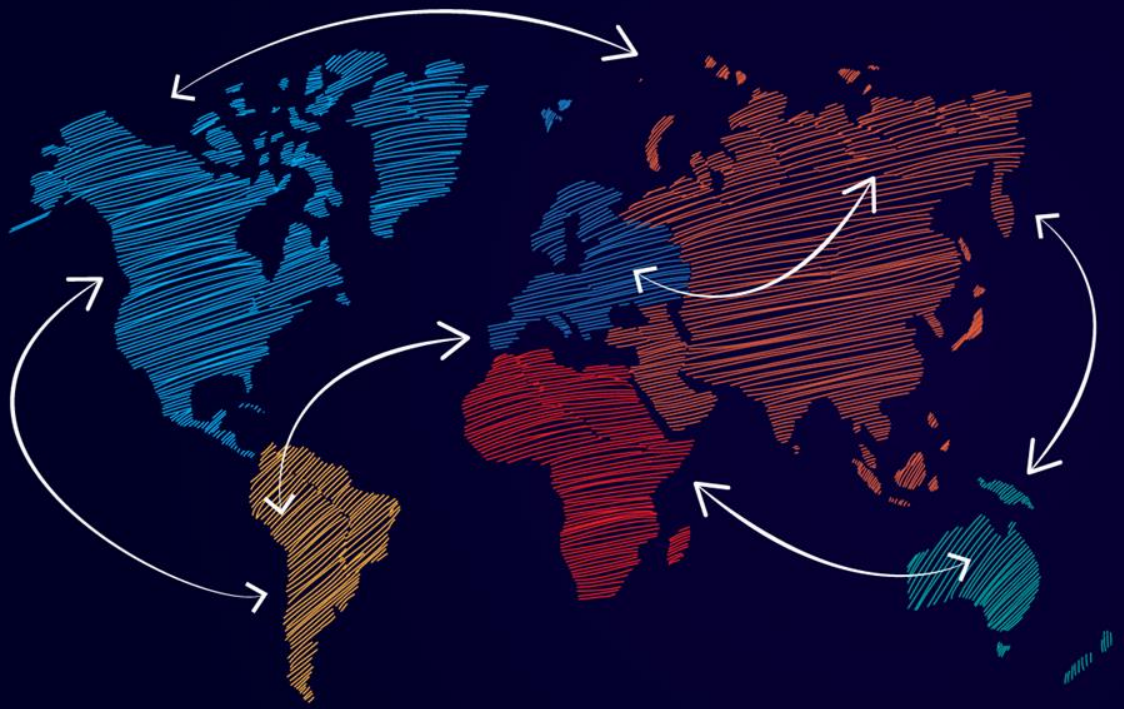
⇒たまたまApexでは最近Barometerの試験は行われていなかった

⇒Apexが試験すれば発覚するが、原因の判断が難しい

※今は修正済み（高橋が実施）

運用面に改善の余地

- 両方ともBarometerのファイル管理の問題ではある
- ゼロベースから試験対応を開始したとき、完璧な対応ができたか？と言われると厳しい
  - 試験Podのパッケージ等は、Barometer内で閉じる話ではない
  - インストーラプロジェクトが叩くテストコードの品質のあるべき姿は？
- **あるべき姿・やりやすい姿の共有は欲しい**
  - **Testingチームからのガイドライン**
  - **Featureチームの経験からのファイル構成のベストプラクティス**



まとめ

OPNFV Testing フレームワークの実体 まとめ

- フレームワークの実装はできている
  - FeatureプロジェクトはFeatureのTest実装に注力できる
  - OPNFV大枠の仕組みにきちんと反映される
- 運用面に改善の余地あり
  - 各プロジェクトのリポジトリの「あるべき姿」を共有したい
  - テスト開発時に何をすればいいのかすぐ理解したい
  - **ドキュメントが欲しい**

参考リンク

- OPNFVのコンポーネントと調べ方
  - https://www.slideshare.net/r-mibu/opnfv-79129688
- OPNFV Jenkins
  - https://build.opnfv.org/ci/
- OPNFV Testing group reporting
  - http://testresults.opnfv.org/reporting/