OpenFog リファレンスアーキテクチャ解説（概要編 – Part II）

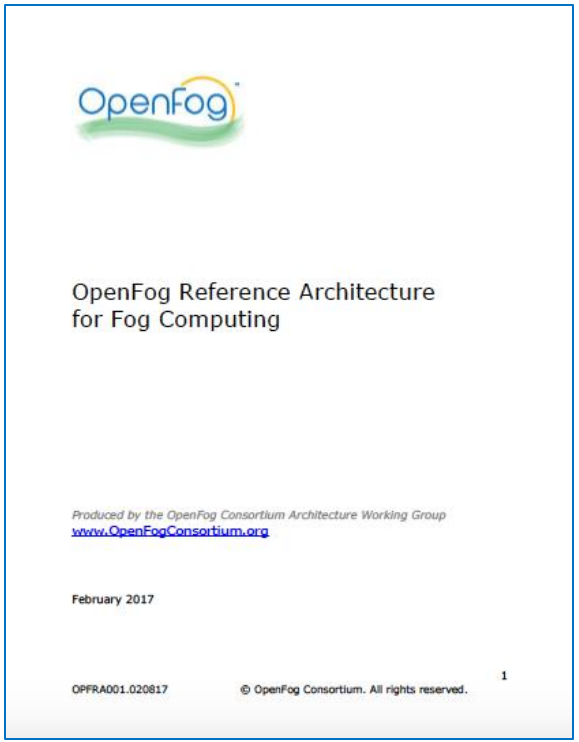
2017年7月21日（金）

OpenFog コンソーシアム 日本地区委員会
シスコシステムズ合同会社　今井俊宏

はじめに

- OpenStackの資料からフォグに関する記述や資料を引用させて頂き、OpenFog Reference Architectureの利用イメージを想定しながら、ご紹介をさせて頂きます。

OpenFog Reference Architecture ドキュメント



- 2017年2月に公開済み
- 以下サイトから基本情報入力だけで、無料でダウンロード可；
  - https://openfog.jp/ra/
- 全11章、約160ページで構成；
  - Chapter 1 About Fog Computing and the Consortium
  - Chapter 2 Areas of Opportunity
  - Chapter 3 Use Cases for Fog
  - Chapter 4 Pillars of OpenFog RA
  - Chapter 5 Reference Architecture Overview
  - Chapter 6 Adherence to OpenFog Reference Architecture
  - Chapter 7 An End-to-End Deployment Use Case
  - Chapter 8 Additional Opportunities
  - Chapter 9 Summary and Next Steps
  - Chapter 10 Appendix –Deeper Security Analysis
  - Chapter 11 Glossary

OpenStack関係者向け
OpenFog Reference Architecture 利用イメージ

openstack

ビジネスリーダー

- 業界の動向を把握する
- フォグコンピューティングに関わるエコシステムの理解を深める
- 様々なフォグコンピューティング利用のユースケースを通じて新しいビジネスアイデアを創出する
- ビジネス戦略の策定や投資戦略へ役立てる

開発者、エンジニア

- フォグコンピューティングを構成する各要素技術を理解する
- ハードウエア、ソフトウエアの側面からフォグコンピューティングの実現方法を理解する
- システムレベルでのフォグコンピューティングの役割を理解する

アカデミア、リサーチャー

- 今後の研究対象領域として理解を深める
- 産業界が直面する最も重要な課題領域として認識を深める
- 進化する技術領域への理解を深めると同時に研究計画への参考にする

等々……

OpenFog

OpenStackでもFogに関する検討されている様なので…

openstack.

Fog Edge Massively Distributed Clouds

The goal of the Fog/Edge/Massively Distributed Clouds working group is to debate and investigate how OpenStack can address Fog/Edge Computing use-cases (i.e. the supervision and use of a large number of remote data centers through a single distributed OpenStack system).

Status: active

Contact: Adrien Lebre <adrien.lebre@inria.fr>

OpenStack WiKiページより抜粋

OpenFog

Use Caseに関しては、こちらの章が参考になると思います；
Chapter 3 Use Cases for Fog

- Chapter 3 構成
  - 3.1 Transportation Scenario: Smart Cars and Traffic Control
  - 3.2 Visual Security and Surveillance Scenario
  - 3,3 Smart Cities Scenario
    - 3.3.1 Smart Building
  - 3.4 Additional Use Cases
- 内容
  - OpenFogアーキテクチャの適用と効果を期待される代表的なUse Caseを解説

Use Caseに関しては、こちらの章が参考になると思います；Chapter 7 An End-to-End Deployment Use Case

- Chapter 7 構成
  - 7.1 Airport Visual Security
  - 7.1.1 Cloud and Edge Approaches
  - 7.1.2 Fog Computing Approach
  - 7.1.3 Application to Airport Visual Security

- 内容
  - OpenFogのアーキテクチャシナリオに照らし合わせて空港利用におけるフォグコンピューチングの利用シーンや利点を考察・解説

Chapter 3 & 7: OpenFogのUse Cases

使用例	 交通システム	 公共安全（監視カメラ）	 スマートシティ/スマートビルディング
社会的な問題	渋滞解消、環境対策、安全な交通網、電気自動車のインフラ整備、今後の普及が見込まれる自動運転への対応等。	犯罪防止、テロ対策、不正侵入や危険物検知等、安心と安全に関わるセキュリティ対策と、セキュリ事故発生時に現場での迅速な判断と対応を行う必要性。	都市におけるビルの安全性、セキュリティ、省電力化、快適性に対する懸念の増大。
システムの課題	個別に構築されるシステムではデータ連携に制約が出る、また、自動運転では、大量に生成されるデータを迅速に処理する必要性がある等。	高画質カメラは、大量のデータを発生。広範囲に敷設される監視カメラ映像を全てクラウドに転送する事は、非効率。プライバシーに対する対策も必須。	ビルに敷設される膨大な数のセンサーから生成される大量のデータを連携させながら、リアルタイム処理する必要性。
フォグによる課題解決	OpenFogアーキテクチャは、自動車や道路等に敷設されるセンサーからの大量のデータを処理、共有、利活用を促進し次世代交通網を支える。	OpenFogアーキテクチャは、クラウドとフォグ間で映像処理を賢く分担し、現場に近い場所でリアルタイムで映像解析を実行、プライバシー保護と同時に、迅速な意思決定とを支える。	OpenFogアーキテクチャは、ビル制御システムに適用し、インテリジェントな空間を創出、また、分散配備されるフォグ連携によりビル監視やビル内制御を支える。

Chapter 3 Use Cases for Fog

- 交通網システムへの適用の可能性を考察

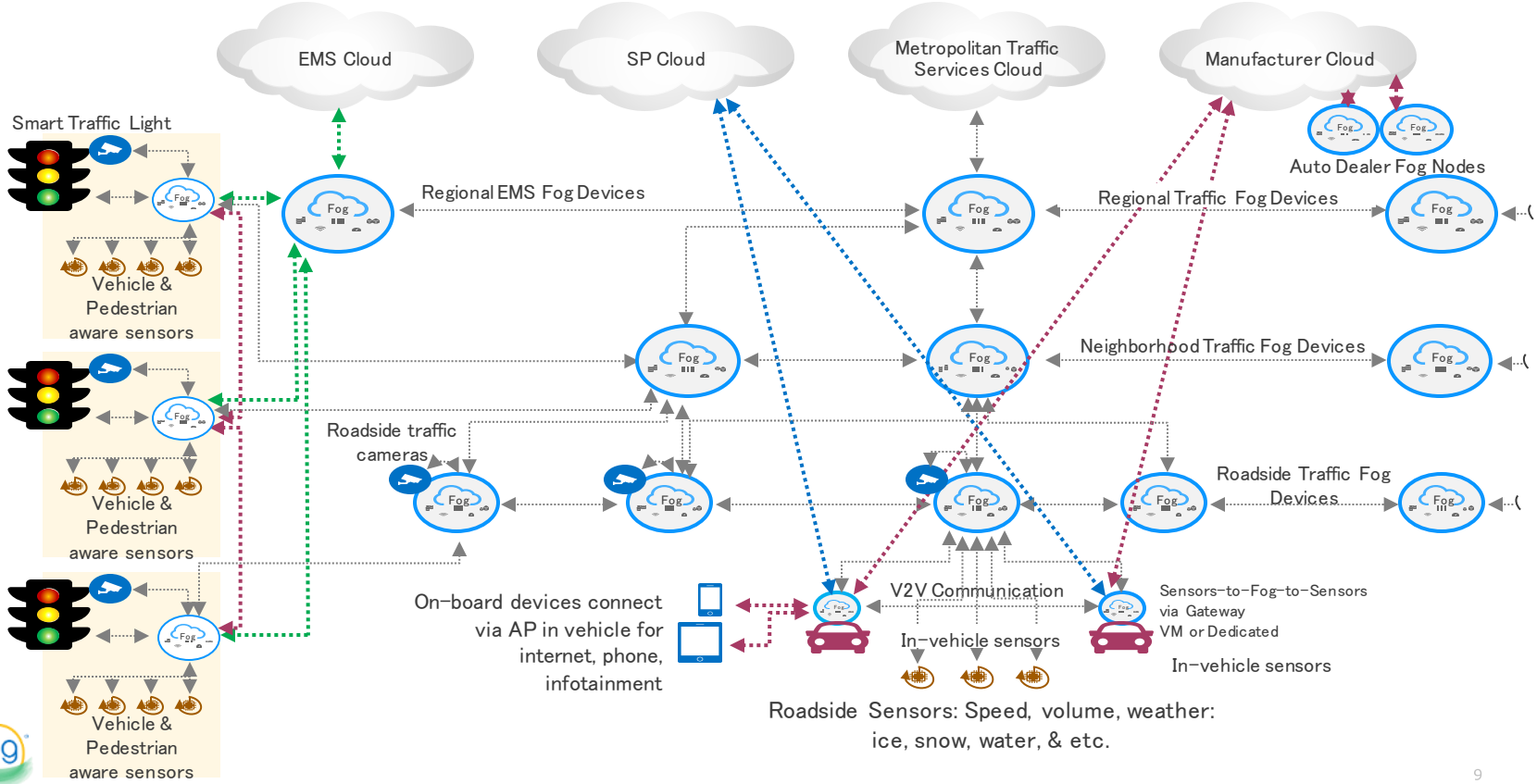


Figure 3 OpenFog Transportation: Smart Car and Traffic Control System

Chapter 3 Use Cases for Fog

- スマートシティへの適用の可能性を考察

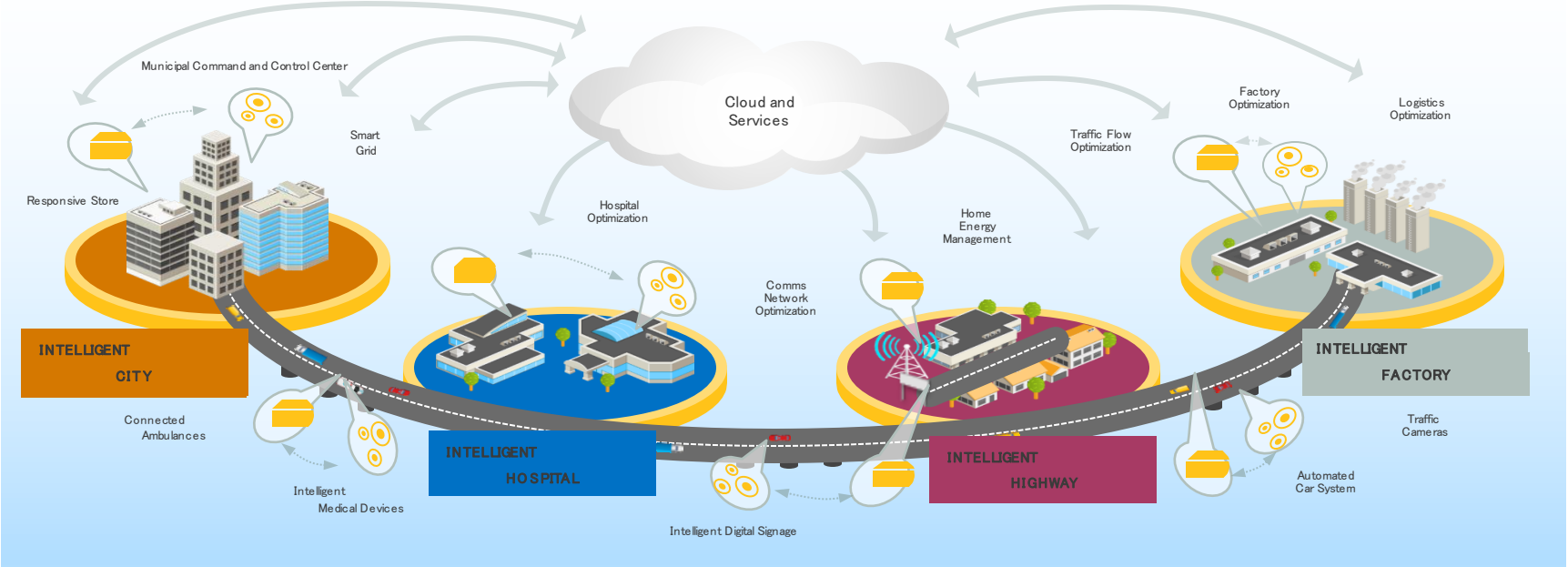


Figure 4 Opportunities for Smart Cities

Chapter 7 An End-to-End Deployment Use Case

- フライトを利用して移動する際、カメラによる画像認識やデータの匿名化等の処理を階層化されたフォグノードの連携で実現するシナリオをOpenFogのアーキテクチャの視点で解説

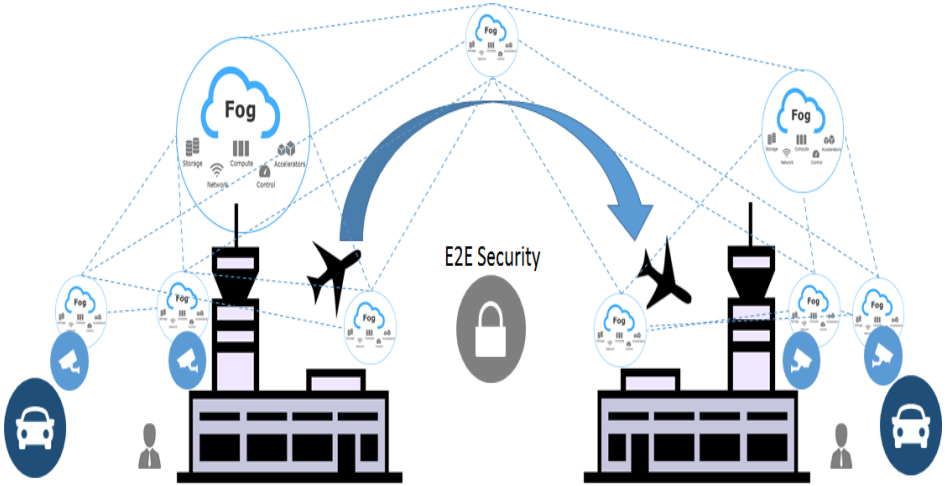


Figure 35 OpenFog realized for Visual Security

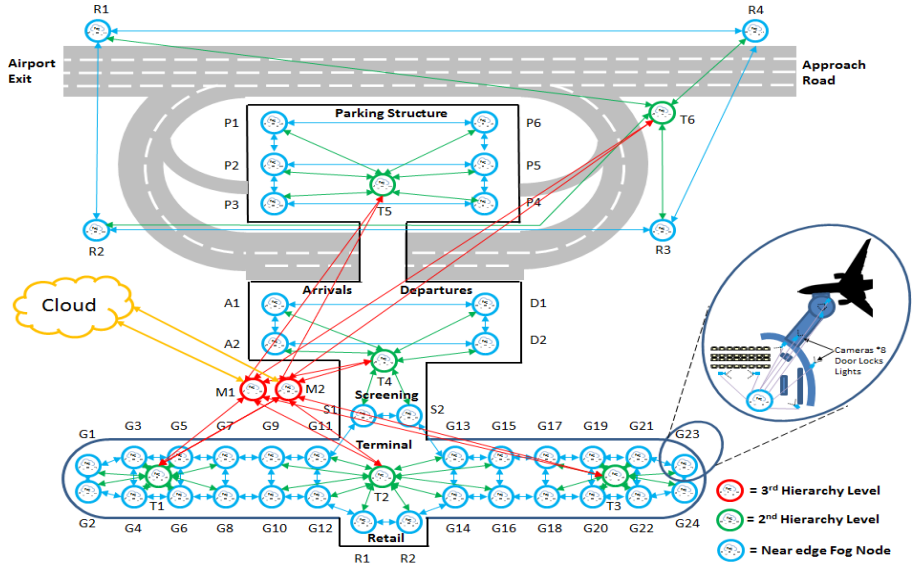
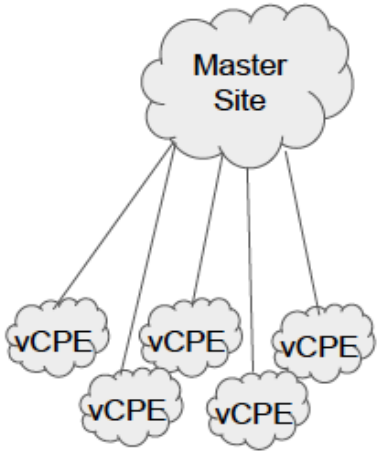


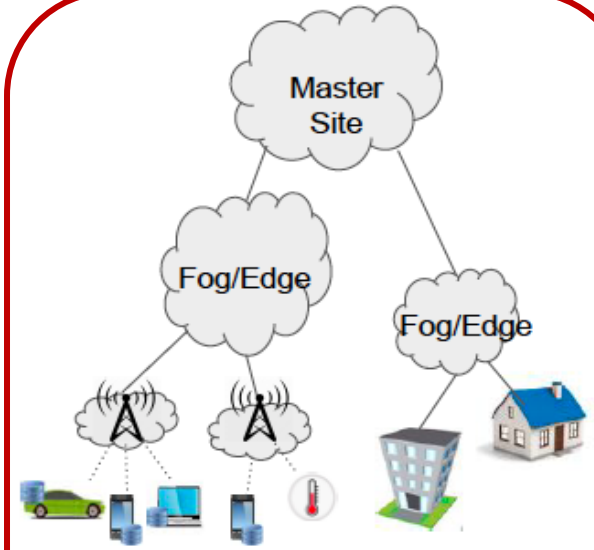
Figure 34 OpenFog Approach to Visual Security Scenario

OpenStackでもFogに関する検討されている様なので…

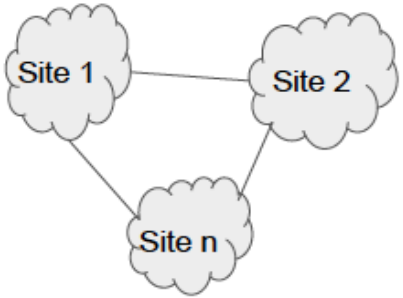
NFV, Fog, Distributed Clouds Use-cases



NFV/SDN
(large scale but provisioning frequency rather low)



Fog/Edge Computing
(Latency aware / smart* / IoT Apps)



Academic cloud federation
(NeCTAR /EGI Federated Cloud)

openstack.

OpenStack資料より抜粋

OpenFog

導入モデルに関しては、こちらの章が参考になると思います；Chapter 4 Pillars of OpenFog RA

- Chapter 4 構成
  - 4.1 Security Pillar
  - 4.2 Scalability Pillar
  - 4.3 Openness Pillar
  - 4.4 Autonomy Pillar
  - 4.5 Programmability Pillar
  - 4.6 Reliability, Availability, and Serviceability (RAS) Pillar
  - 4.7 Agility Pillar
  - 4.8 Hierarchy Pillar
  - 4.9 Hierarchical Fog Deployment Models
- 内容
  - OpenFogアーキテクチャーを支える8つの主要素（Pilar）とフォグの導入モデルに関して、フォグのエコシステム（部品製造、システムベンダー、ソフトウエア開発、アプリケーション開発等）向けに解説

Chapter 4 Pillars of OpenFog RA

- IoTシステム導入モデル比較

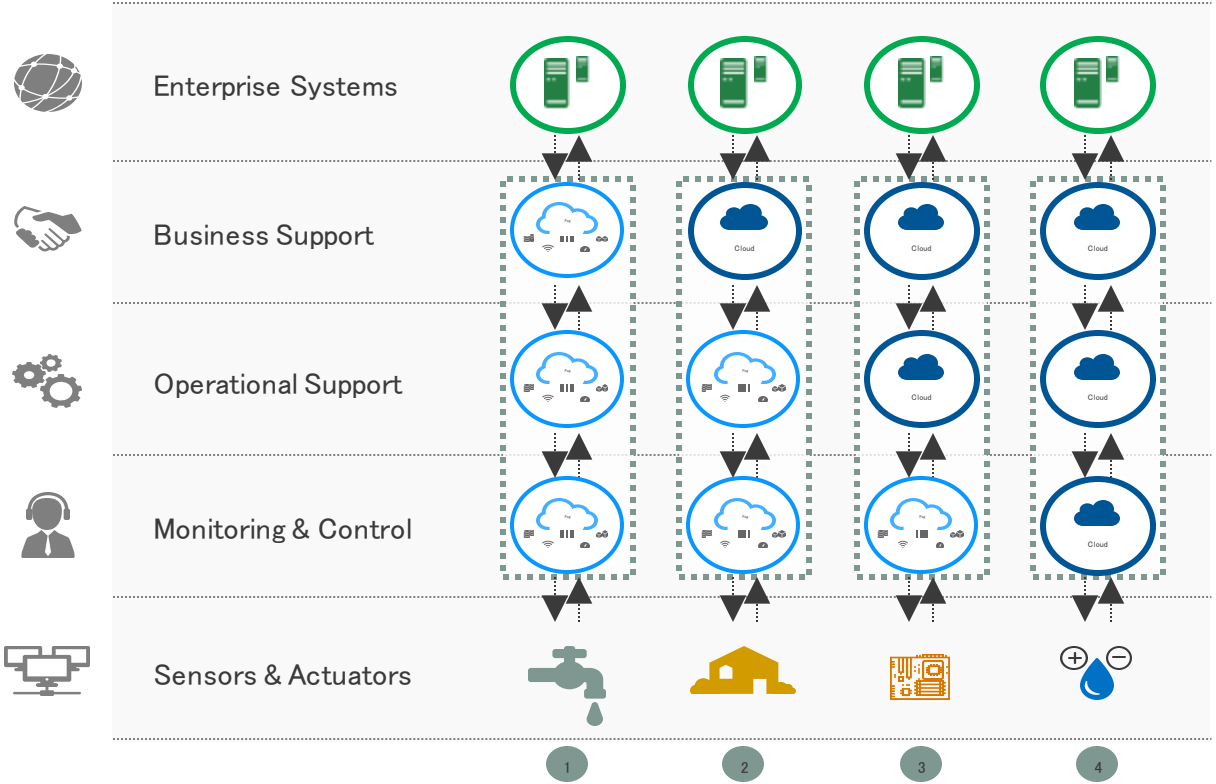


Figure 7 IoT System Deployment Models

Chapter 4 Pillars of OpenFog RA

- 階層型フォグ構成の例

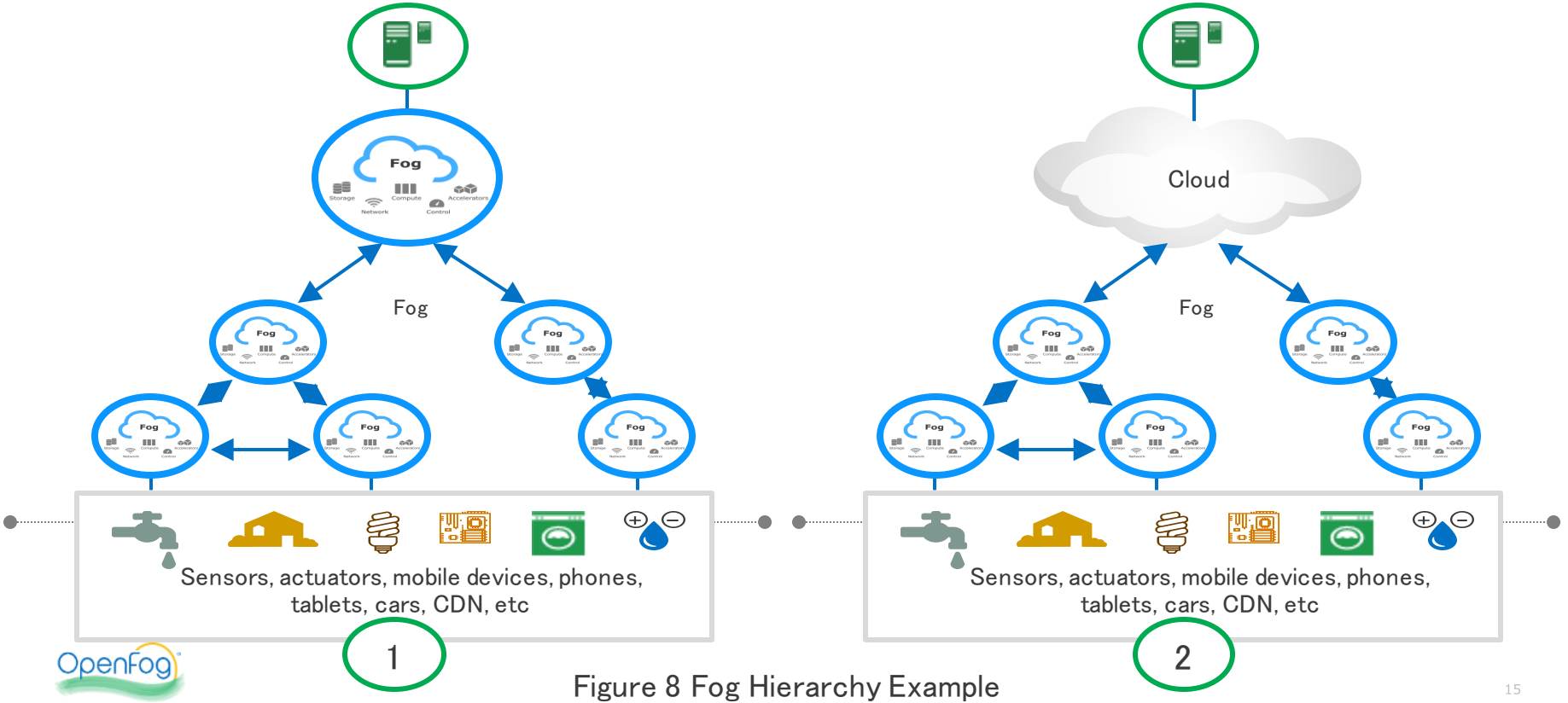


Figure 8 Fog Hierarchy Example

Chapter 4 Pillars of OpenFog RA

- 階層型フォグ構成の例

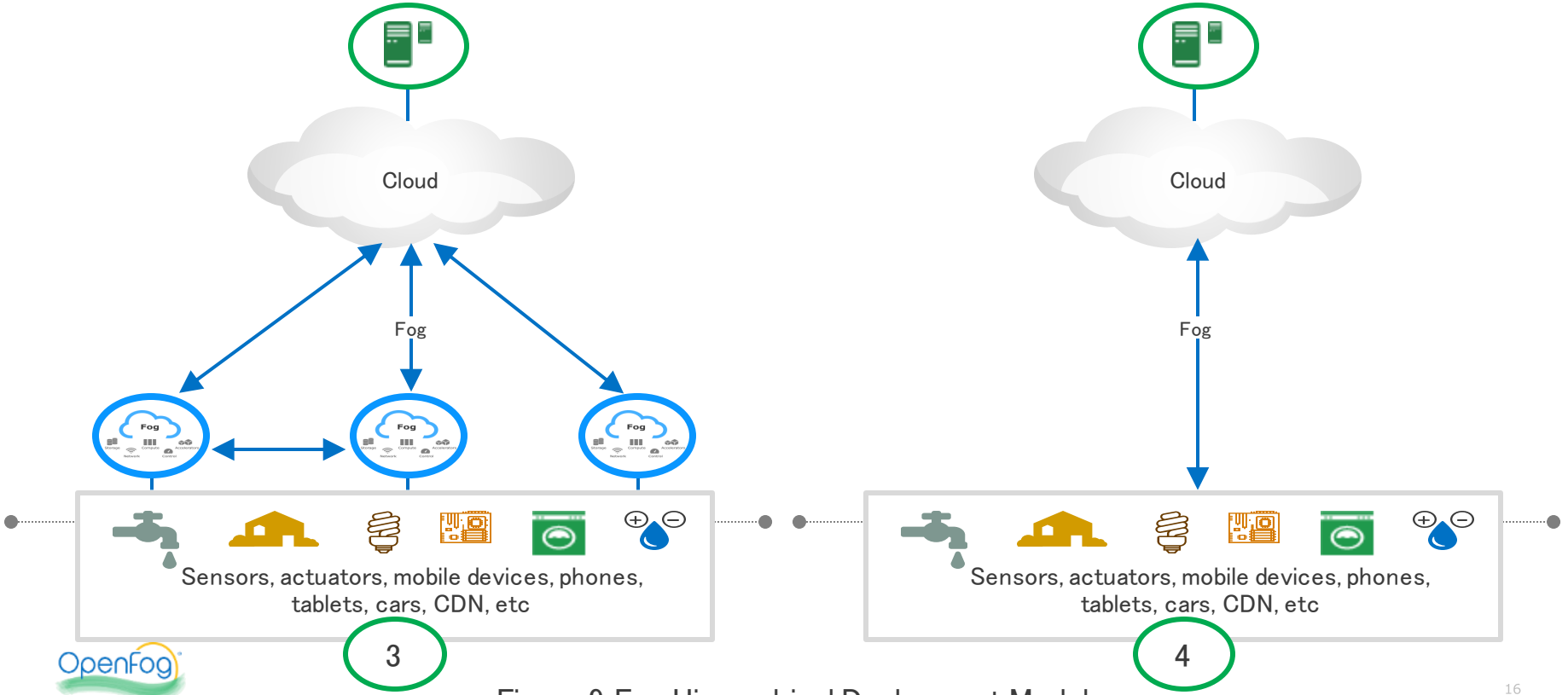


Figure 9 Fog Hierarchical Deployment Model

F/E/MDC WGのProblem Statementから…

openstack

- Scalability of the controller: How many controller should/could be deployed to supervise the whole infrastructure? on which location(s)? One per site, one for several sites? How many compute nodes per controller would be necessary?
- Should we have a single or multiple endpoints? Why?
- Wide Area Network limitations (in terms of latency/bandwidth): Are there critical latency constraints that may prevent the correct functioning of core components? Are current services efficient enough to deal with WAN constraints (VM images, …)
- Consistency: How can we guarantee consistency of core-services states? If one project/vm/… is created on one site, the states of the other sites should be consistent to avoid for instance double assignment of Ids/IPs/…
- Security management : Do Fog/Edge infrastructure create new security issues ? How can we ensure the security of communications inside and between the different locations?
- Fault tolerance issues: How can we revise OpenStack in a way that guarantees that the crash or the isolation of one (or several sites) does not impact other DCs? (Each site should be able to run independently.)
- Maintainability: how can we upgrade the system in a consistent way (considering that upgrading the complete infrastructure can take a significant amount of time while facing crash and disconnection issues) ? In other words, we should propose mechanisms that allow OpenStack to behave correctly even if we have different versions of the core-services?
- Interconnexion between multi-vendors (peering agreement challenges, interoperability…)

OpenFog

OpenStack WiKiページより抜粋

Problem Statementには、こちらの章が参考になると思います；
Chapter 5 Reference Architecture Overview

- Chapter 5 構成
  - 5.1 Functional Viewpoint
  - 5.2 Deployment Viewpoint
    - 5.2.1 OpenFog Deployment Types
    - 5.2.2 N-Tier Fog Deployment
  - 5.3 OpenFog Architecture Description
  - 5.4 Perspectives (Cross Cutting Concerns)
    - 5.4.1 Performance and Scale Perspective
    - 5.4.2 Security Perspective
    - 5.4.3 Manageability Perspective
    - 5.4.4 Data, Analytics, and Control
    - 5.4.5 IT Business and Cross-fog Applications

Chapter 5 Reference Architecture Overview

5.5 Node View
- 5.5.1 Network
- 5.5.2 Accelerators
- 5.5.3 Compute
- 5.5.4 Storage
- 5.5.5 OpenFog Node Management
- 5.5.6 OpenFog Node Security

5.6 System Architecture View
- 5.6.1 Hardware Platform Infrastructure
- 5.6.2 Hardware Virtualization and Containers

5.7 Software Architecture View
- 5.7.1 Software View Layers

- 内容

  リファレンスアーキテクチャーのコアとなる各構成要素に関して解説

F/E/MDC WGのProblem Statementから…

openstack.

- Scalability of the controller: How many controller should/could be deployed to supervise the whole infrastructure? on which location(s)? One per site, one for several sites? How many compute nodes per controller would be necessary?
- Should we have a single or multiple endpoints? Why?
- Wide Area Network limitations (in terms of latency/bandwidth): Are there critical latency constraints that may prevent the correct functioning of core components? Are current services efficient enough to deal with WAN constraints (VM images, …)
- Consistency: How can we guarantee consistency of core-services states? If one project/vm/… is created on one site, the states of the other sites should be consistent to avoid for instance double assignment of Ids/IPs/…
- Security management Do Fog/Edge infrastructure create new security issues ? How can we ensure the security of communications inside and between the different locations?
- Fault tolerance issues: How can we revise OpenStack in a way that guarantees that the crash or the isolation of one (or several sites) does not impact other DCs? (Each site should be able to run independently.)
- Maintainability: how can we upgrade the system in a consistent way (considering that upgrading the complete infrastructure can take a significant amount of time while facing crash and disconnection issues) ? In other words, we should propose mechanisms that allow OpenStack to behave correctly even if we have different versions of the core-services?
- Interconnexion between multi-vendors (peering agreement challenges, interoperability…)

OpenFog

OpenStack WiKiページより抜粋

FogのSecurityに関しては、こちらの章も参考になると思います；Chapter 10 Appendix – Deeper Security Analysis

- Chapter 10 構成
  - 10.1 Security Aspects
    - 10.1.1 Cryptographic Functions
    - 10.1.2 Node Security Aspect
    - 10.1.3 Network Security Aspect
    - 10.1.4 Data Security Aspect
- 内容
  - OpenFogのリファレンスアーキテクチャのおけるセキュリティを深掘りAppendixとして考察・解説

F/E/MDC WGのProblem Statementから…

openstack.

- Scalability of the controller: How many controller should/could be deployed to supervise the whole infrastructure? on which location(s)? One per site, one for several sites? How many compute nodes per controller would be necessary?
- Should we have a single or multiple endpoints? Why?
- Wide Area Network limitations (in terms of latency/bandwidth): Are there critical latency constraints that may prevent the correct functioning of core components? Are current services efficient enough to deal with WAN constraints (VM images, …)
- Consistency: How can we guarantee consistency of core-services states? If one project/vm/… is created on one site, the states of the other sites should be consistent to avoid for instance double assignment of Ids/IPs/…
- Security management : Do Fog/Edge infrastructure create new security issues ? How can we ensure the security of communications inside and between the different locations?
- Fault tolerance issues: How can we revise OpenStack in a way that guarantees that the crash or the isolation of one (or several sites) does not impact other DCs? (Each site should be able to run independently.)
- Maintainability: how can we upgrade the system in a consistent way (considering that upgrading the complete infrastructure can take a significant amount of time while facing crash and disconnection issues) ? In other words, we should propose mechanisms that allow OpenStack to behave correctly even if we have different versions of the core-services?
- Interconnexion between multi-vendors (peering agreement challenges, interoperability…)

OpenFog

OpenStack WiKiページより抜粋

フォグの管理概念に関する解説が参考になると思います; Chapter 5 Reference Architecture Overview

- フォグノードのライフサイクル管理

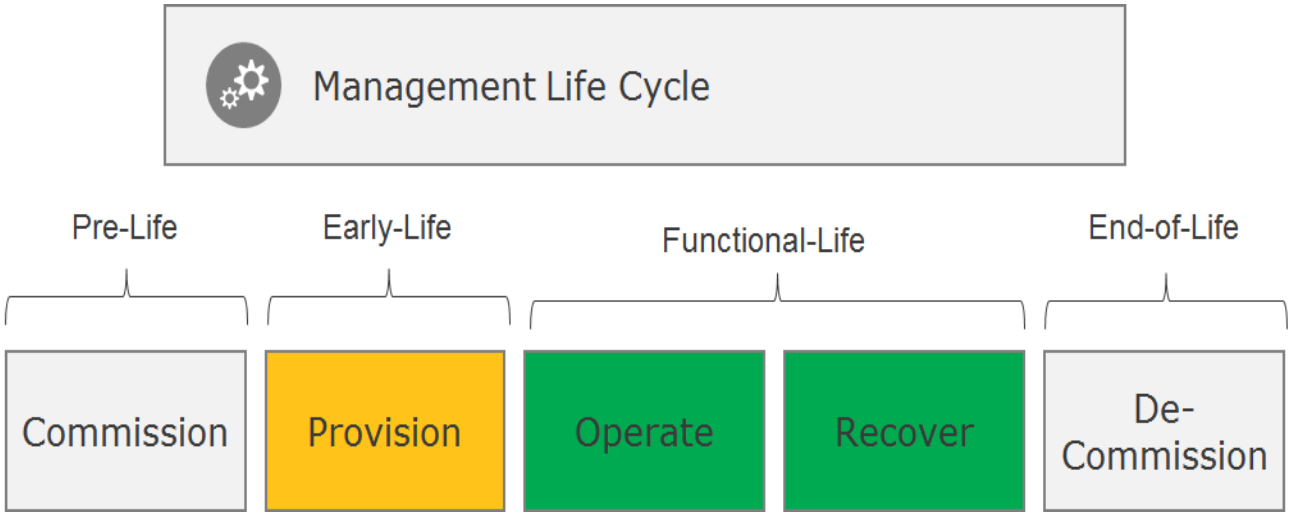


Figure 17 Management Life Cycle

Chapter 5 Reference Architecture Overview

- フォグノードの管理レイヤー

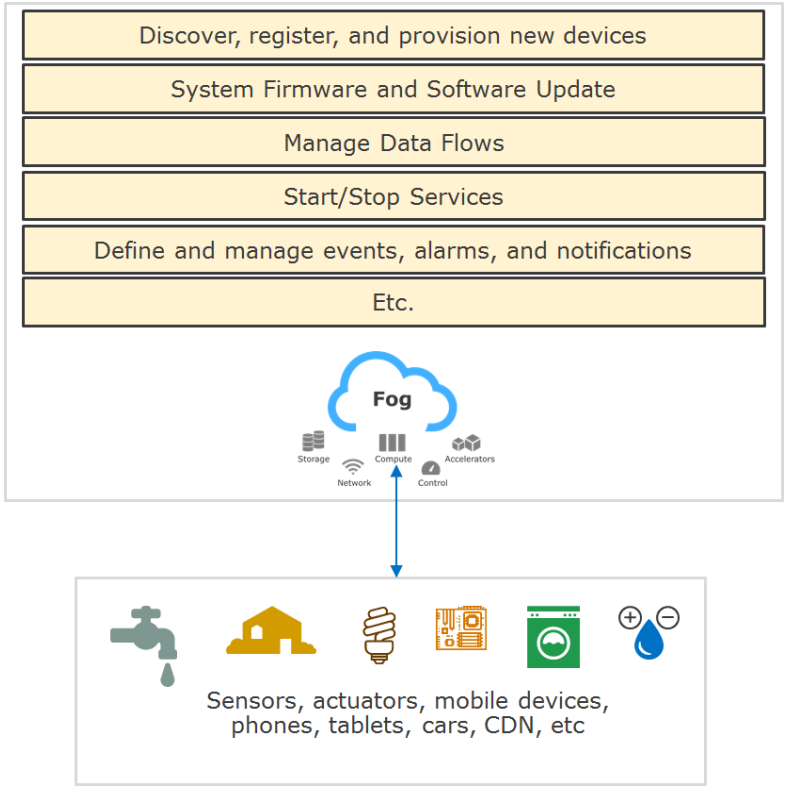


Figure 18 Management Layer

F/E/MDC WGのProblem Statementから…

openstack.

- Scalability of the controller: How many controller should/could be deployed to supervise the whole infrastructure? on which location(s)? One per site, one for several sites? How many compute nodes per controller would be necessary?
- Should we have a single or multiple endpoints? Why?
- Wide Area Network limitations (in terms of latency/bandwidth): Are there critical latency constraints that may prevent the correct functioning of core components? Are current services efficient enough to deal with WAN constraints (VM images, …)
- Consistency: How can we guarantee consistency of core-services states? If one project/vm/… is created on one site, the states of the other sites should be consistent to avoid for instance double assignment of Ids/IPs/…
- Security management : Do Fog/Edge infrastructure create new security issues ? How can we ensure the security of communications inside and between the different locations?
- Fault tolerance issues: How can we revise OpenStack in a way that guarantees that the crash or the isolation of one (or several sites) does not impact other DCs? (Each site should be able to run independently.)
- Maintainability: how can we upgrade the system in a consistent way (considering that upgrading the complete infrastructure can take a significant amount of time while facing crash and disconnection issues) ? In other words, we should propose mechanisms that allow OpenStack to behave correctly even if we have different versions of the core-services?
- Interconnexion between multi-vendors (peering agreement challenges, interoperability…)

OpenFog

OpenStack WiKiページより抜粋

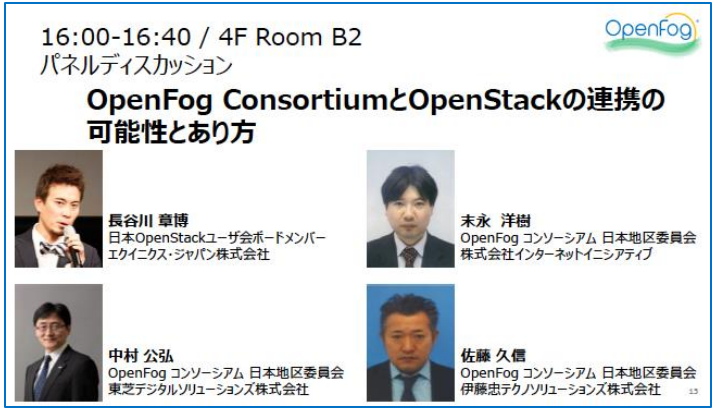
相互接続性の取組みは、こちらの章が参考になると思います；Chapter 6 Adherence to OpenFog Architecture

- 内容
  - 標準化への指針に言及、また、テストベッドにおける相互接続性検証や認証に関するする必要性を解説

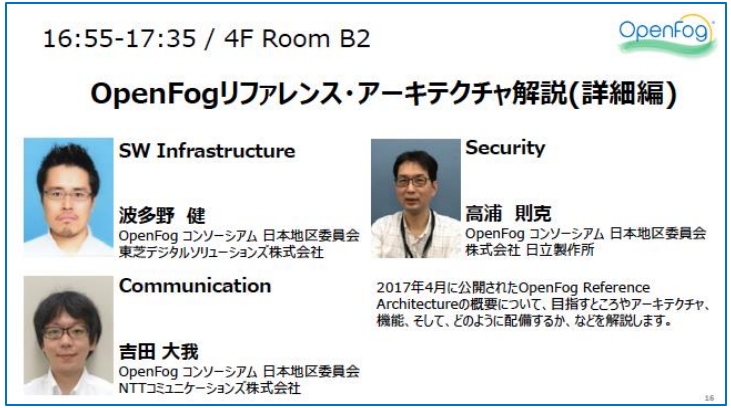
Figure 28 OpenFog Technology Ready

Figure 29 OpenFog Ready

おわりに



OpenFog RAの要素技術の詳細や、
日本発で検討が進むUse Case、
OpenStackとの連携話は、
引続きのこちらの各セッションを聴講下さい。




OpenFog
www.OpenFogConsortium.org
日本語サイト：https://openfog.jp