HITACHI
Inspire the Next

Edge computing with Node-RED

Yu Nakata

HITACHI
Inspire the Next

Introduction

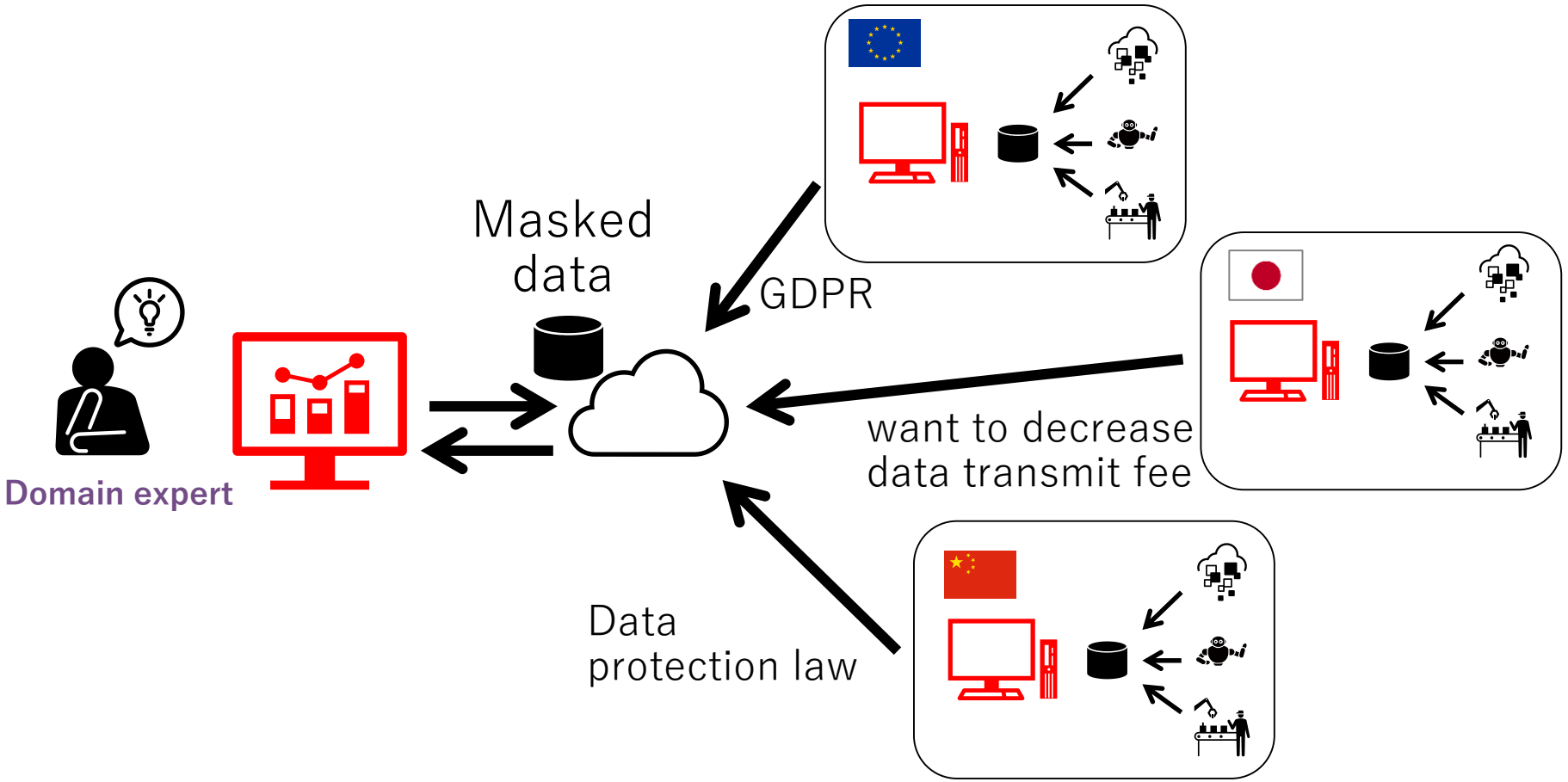
Our approach

Discussion

Edge computing

HITACHI
Inspire the Next

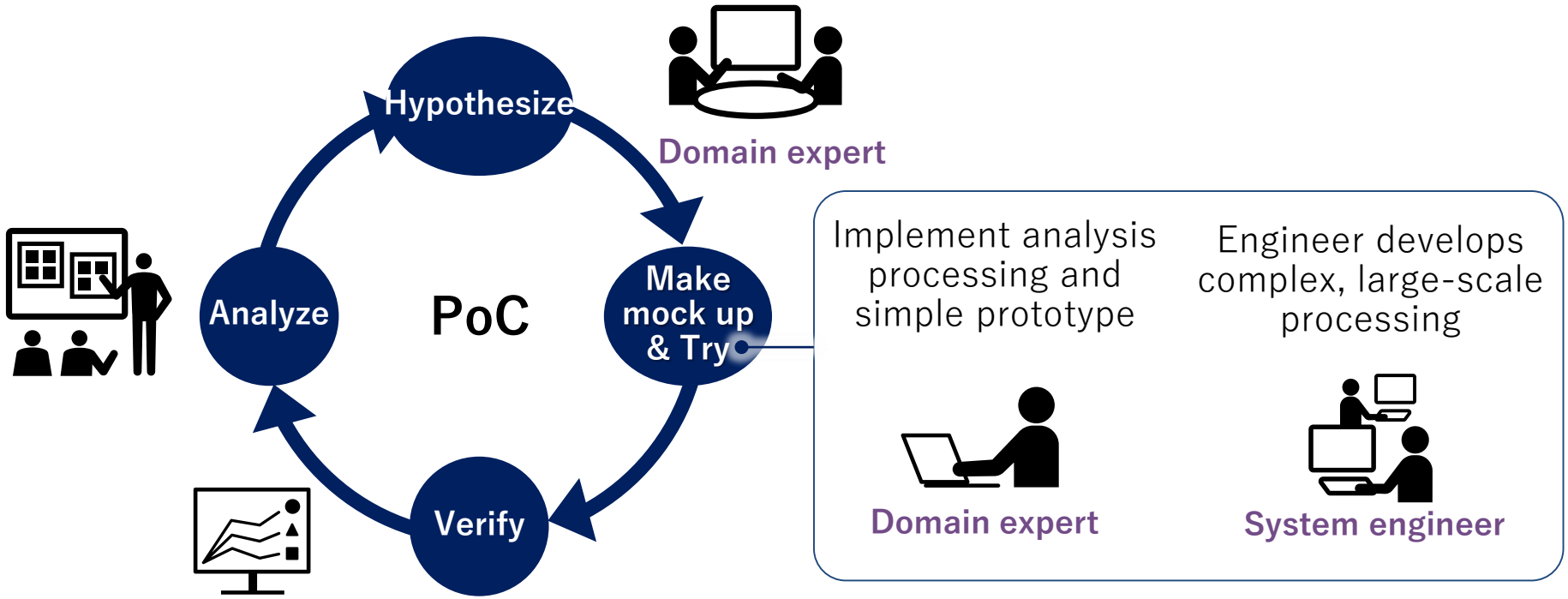
- A form of distributed processing that performs data processing locally
  - to process data with low latency
  - to process data that cannot to be on cloud (e.g., sensitive data)



System developing process in SoE (System of Engagement)

HITACHI
Inspire the Next

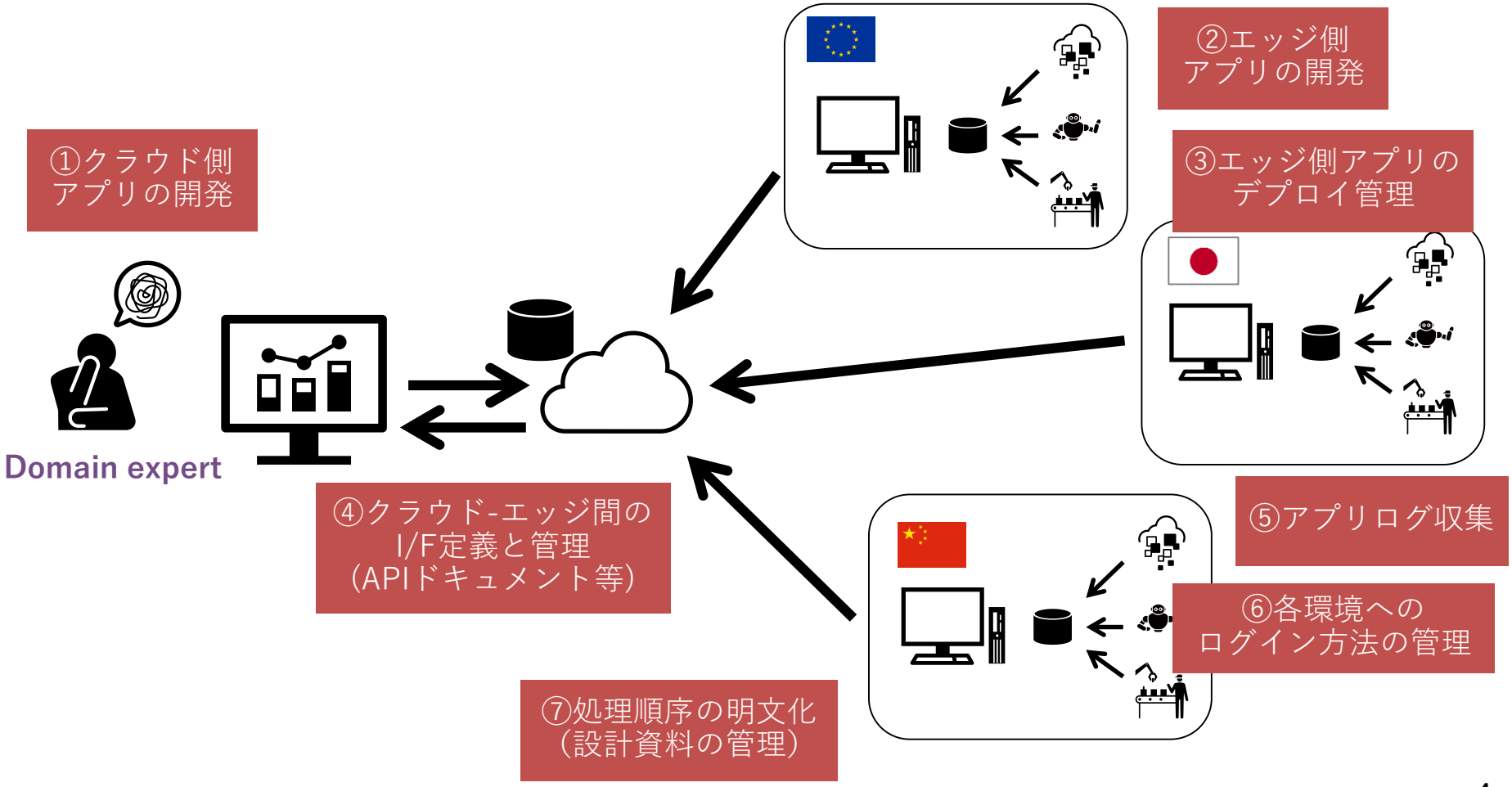
- Important point is utilizing data that depict activities, status and events of human life, business, environment and so on
- Only domain experts can really understand the meaning of the data
- It is critical that domain experts participate in PoC
  - However, most domain experts have less programming experience



Challenge of edge computing for domain experts

HITACHI
Inspire the Next

- There are a lot of tasks that app developers must do in application development, execution, testing and management



Our approach

HITACHI
Inspire the Next

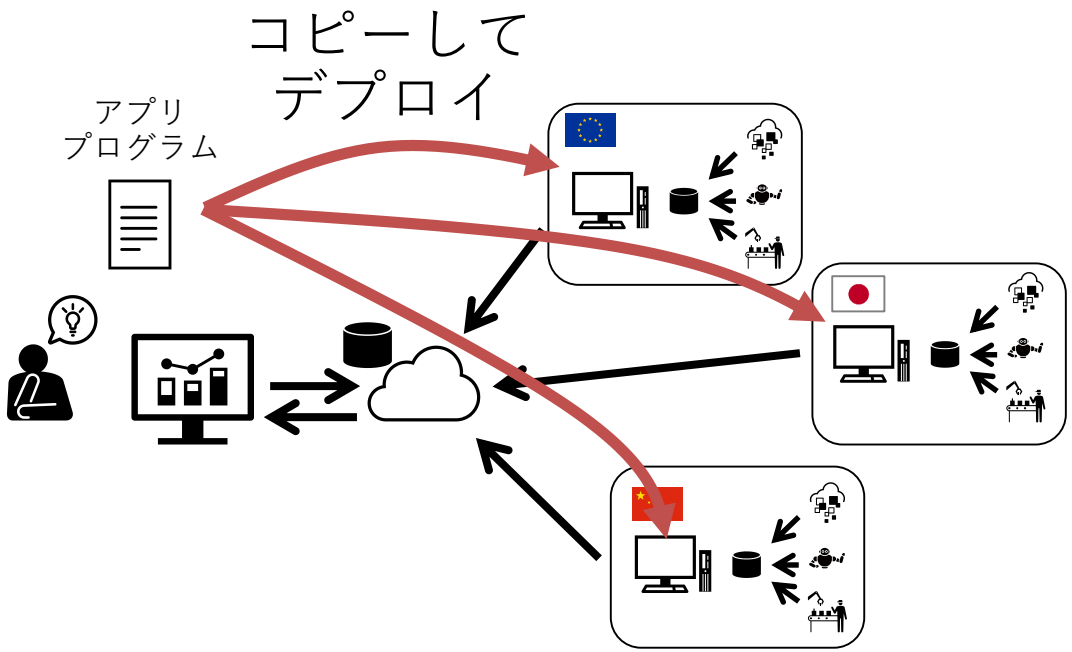
Root cause of problem

１つのソリューションが
<u>複数のアプリで実現</u>され、
それぞれが
異なる場所で稼動すること

Approach

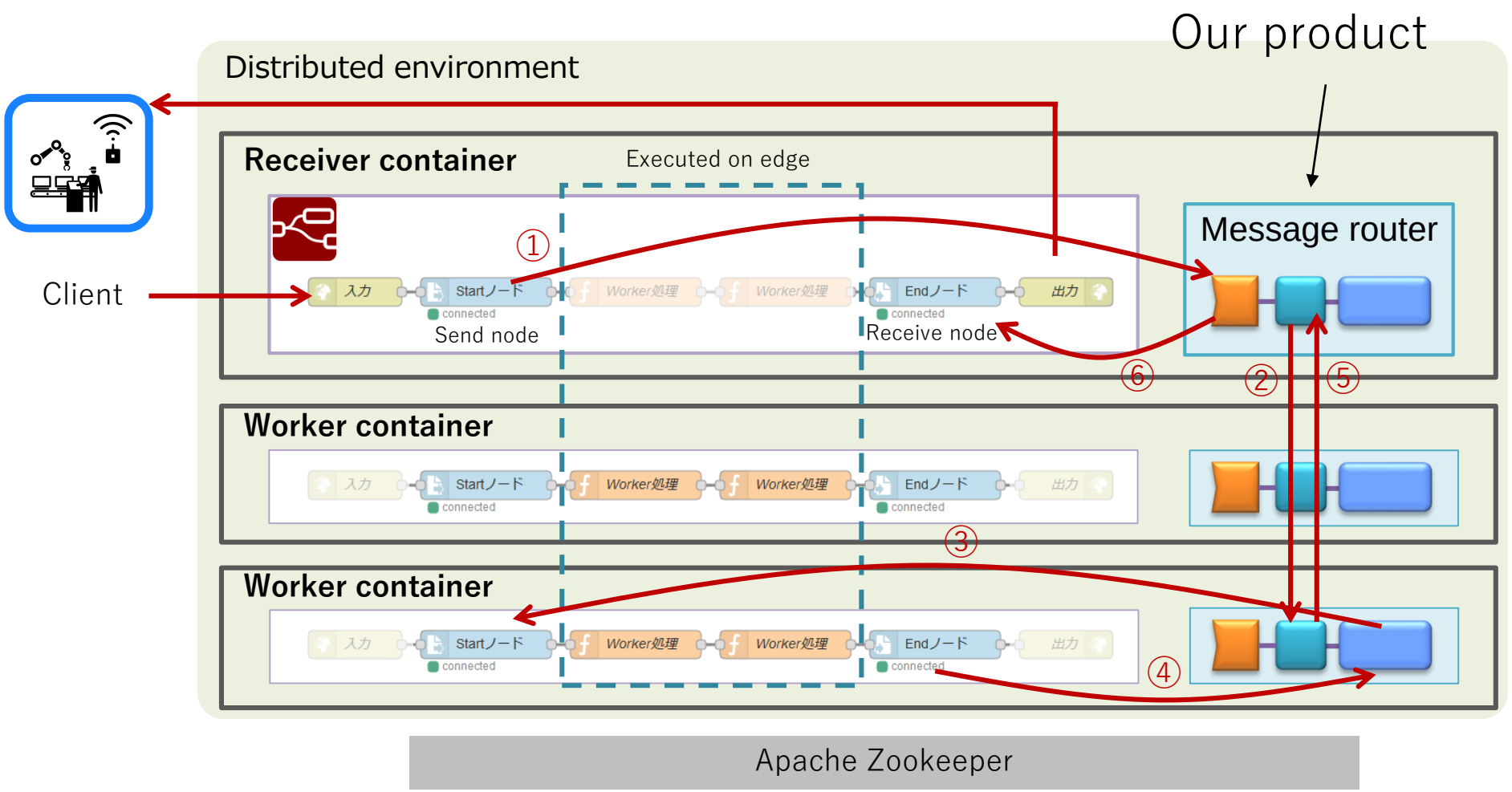
１つのソリューションを
<u>１つのアプリで実現</u>する

同じアプリを、
複数の場所にデプロイする



Structure of execution environment

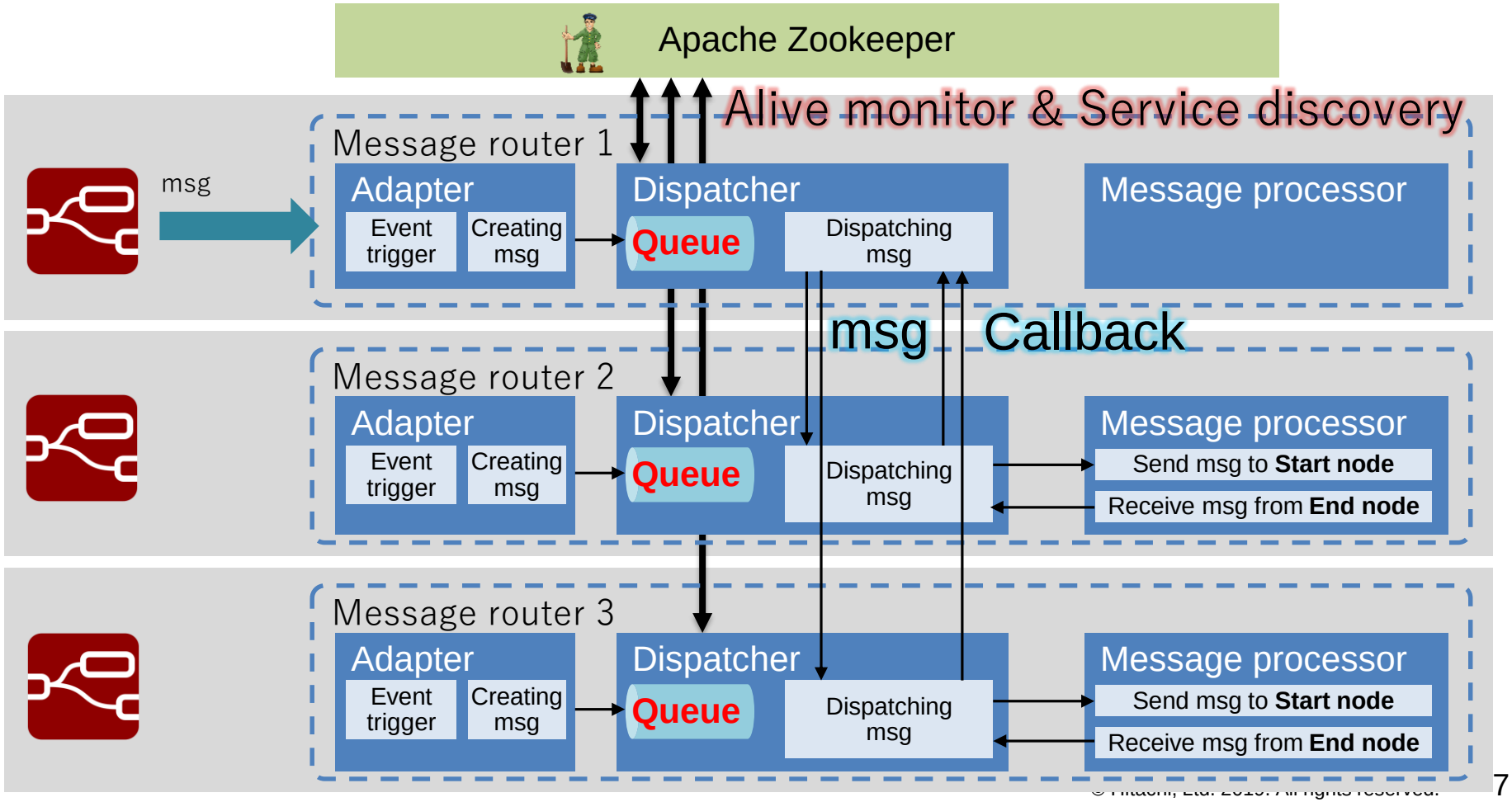
- Message routerのDispatcherがZookeeperから拠点情報を取得
- Dispatcherが、メッセージ内で指定された拠点に、メッセージを転送



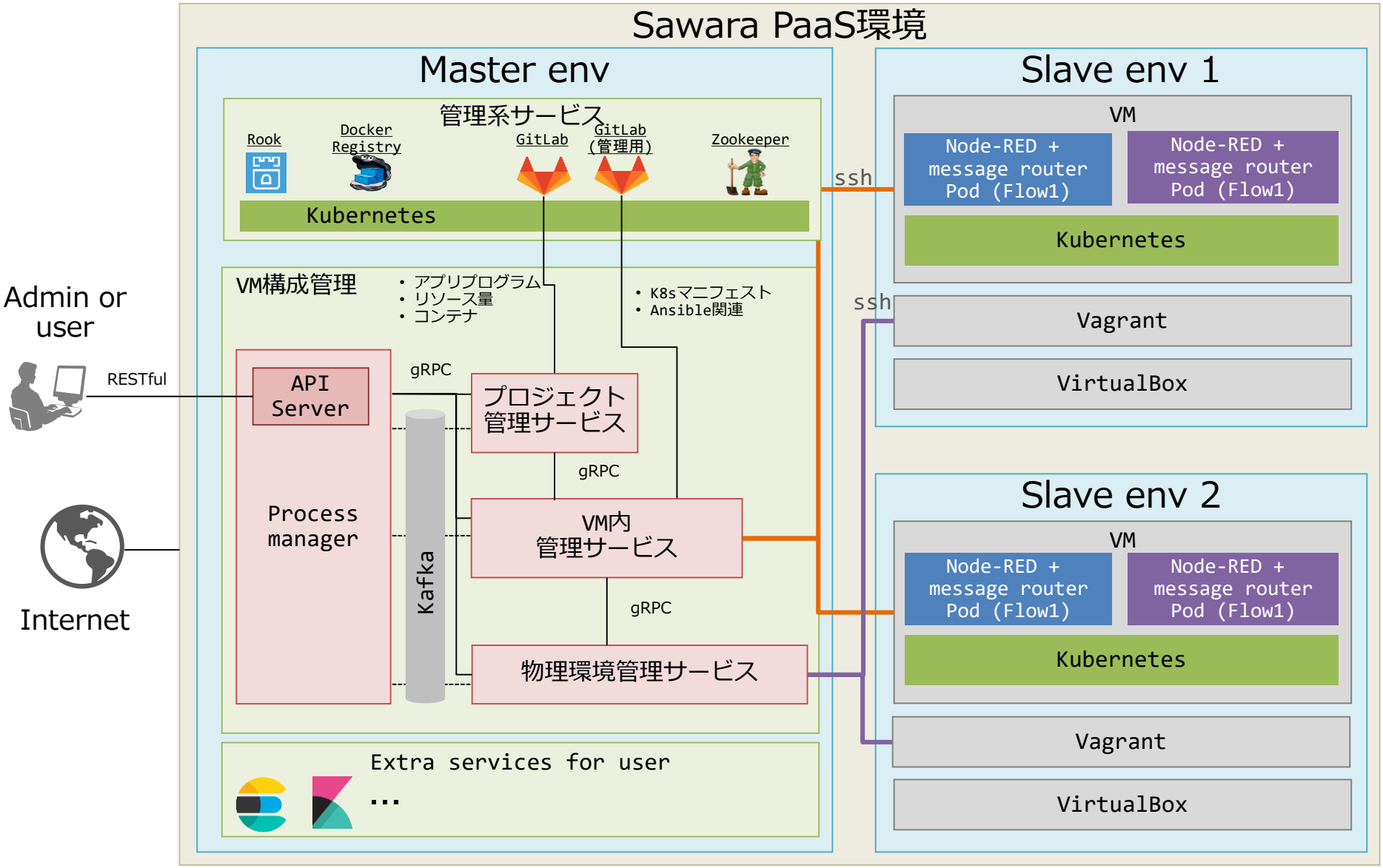
Message router

非同期型のメッセージパッシングで、ノンブロッキングに処理分散

- 高信頼（キューにメッセージを保持し、障害時には別サーバーに自動再送）
- 拡張性（クラスタ内のコンテナを自動検知し、動的にスケール可能）



Location attribute on VM, container



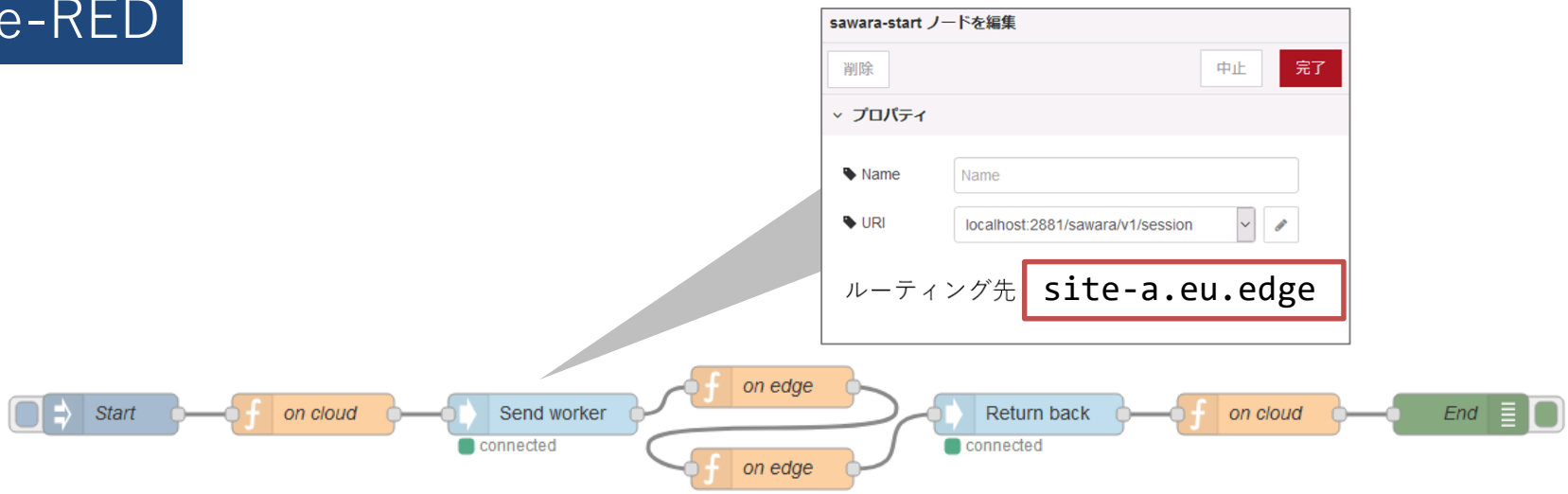
自作サービス　ファイル

Sawara PaaS環境内の全k8sノードは同一のクラスタに所属

User experience of development

HITACHI
Inspire the Next

Node-RED



Python

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from sawara import sawara

@sawara.sawara_paas(dist="site-a.eu.edge")
def greet_morning(*args):
    print('Good morning!')

def main():
    greet_morning()
    sawara.close()

if __name__ == '__main__':
    main()
```

このデコレータをつけると、
その下の関数のみ
別の環境で実行される

`site-a.eu.edge`とラベル付けされた環境で実行

Discussion

HITACHI
Inspire the Next

We found some distributed flow execution approaches
Main two approaches
1. **Deploying partial flows/nodes approach** [1,2]
2. Message router approach [3,4] (*c.f. Enterprise integration patterns*)

> “ The participating D-NR (Distributed Node-RED) instances then parse the *master flow* and based on a set of constraints, **decide which nodes should to be deployed locally and which are to be replaced by a *placeholder* node**. The *placeholder* node is used to connect *sub flows* from different devices together.

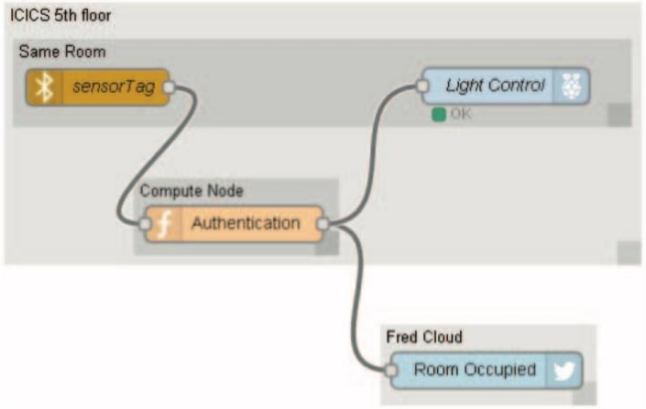


Fig. 4. Develop Fog-based IoT application with D-NR

[1] N. K. Giang and et al., "Developing IoT Applications in the Fog: a Distributed Dataflow Approach ," in proceedings of the 5th International Conference on the Internet of Things (IoT), pp. 155-162, Oct. 2015.
[2] J. Hosie and et al., "Secure message handling of an application across deployment locations," United States Patent, Sep. 2016 (filed).

Discussion

We found some distributed flow execution approaches
Main two approaches
1. Deploying partial flows/nodes approach [1,2]
2. **Message router approach** [3] (***c.f. Enterprise integration patterns***)
   => like *pluggable message routing*

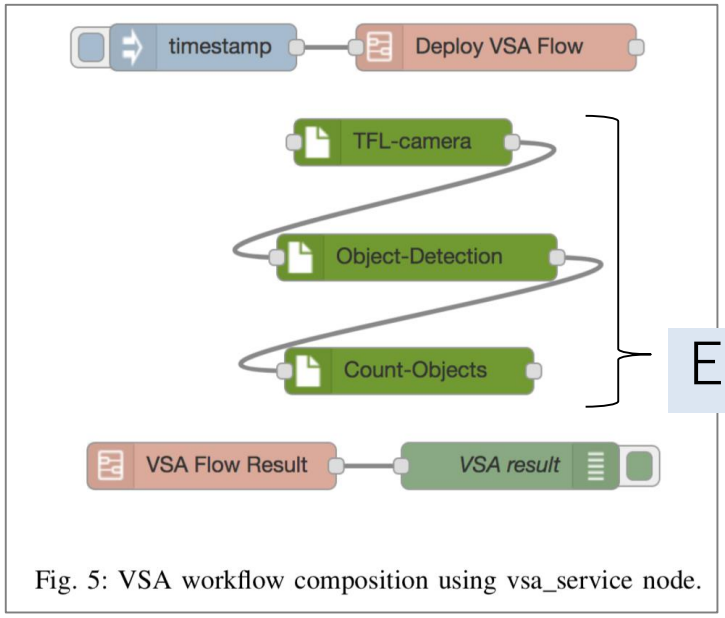


Fig. 5: VSA workflow composition using vsa_service node.

Executed on edge

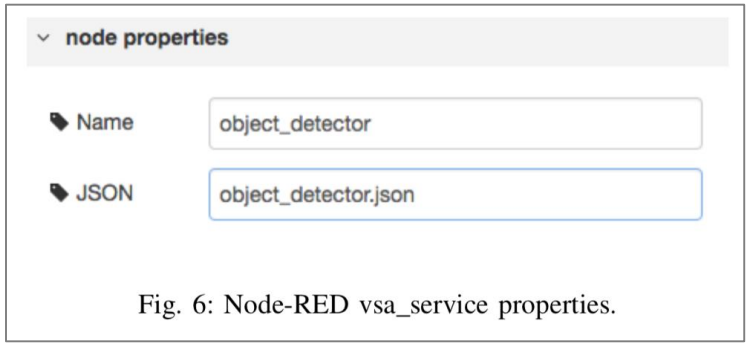


Fig. 6: Node-RED vsa_service properties.

[3] C. Simpkin and et al., "Dynamic Distributed Orchestration of Node-RED IOT Workflows Using a Vector Symbolic Architecture," in Proceedings of the 13th Workshop on Workflows in Support of Large-Scale Archive Listing, Oct. 2018.

Discussion

HITACHI
Inspire the Next

- Since needs for distributed flow execution is growing, implement interfaces supporting the function is highly valuable
  - We also would like to follow the interfaces
  - How high is a priority of distributed flow execution
- Some approaches have already proposed
  - How do you organize these approaches?
  - (We'd prefer *message router approach*, so if there are discussion on anywhere, we'd like to participate in it.)