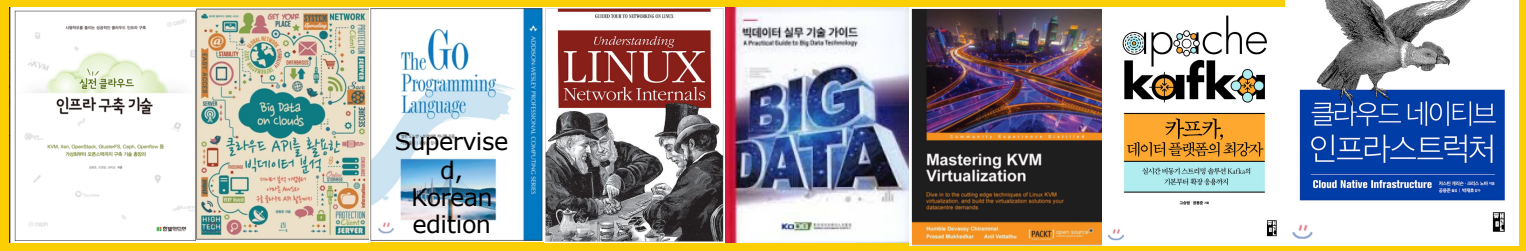
cloud ,
cloud Native
cloud Hybrid

공용준(Andrew.kong)
kakao corp

Who am I

Andrew. Yongjoon kong

- Cloud Technical Advisory for Government Broad Cast Agency
- Adjunct Prof. Ajou Univ
- Korea Data Base Agency Acting Professor for Bigdata
- Member of National Information Agency Bigdata Advisory committee
- Kakaocorp, Cloud Tech Director
- Talks
  - cloud native platform 9rum (2018, if kakao, Korea)
  - Scalable Loadbalancer with VM orchestrator (2017, netdev, Korea)
  - Embrace clouds (2017, openstack days, Korea)
  - Algorithmic Economy with cloud (2016,IDG, Korea)
  - Full route based network with linux (2016, netdev, Tokyo)
  - SDN without SDN (2015, openstack, Vancouber)



Before start - 0

- 자기 소개 하는 시간
  - 이름
  - 하고 계시는 일
  - 여기에 온 저의(?) 등

Before start

- Install virtualbox (https://www.virtualbox.org/wiki/Downloads)
- Install vagrant (www.vagrantup.com → downloads 2.2.4)
- download Vagrantfile (https://bit.ly/2KHDTxd)
- go to directory where Vagrant file exists
- ‘vagrant up’

First of All

- What is Cloud?

What is Cloud Computing?

Kakao workshop, 2014

What we are.

Cloud Computing =
Programmable Resource Managing

What is Programmable?

Kakao workshop, 2014

Terms

- Programmable Resource Management
  - something standardized (based on policy, rules)
  - something abstracted (behind the scene)
  - something automated (not manual job)
  - something repeated regularly
  - ‘programmable’ always implies certain level

What is Resource?

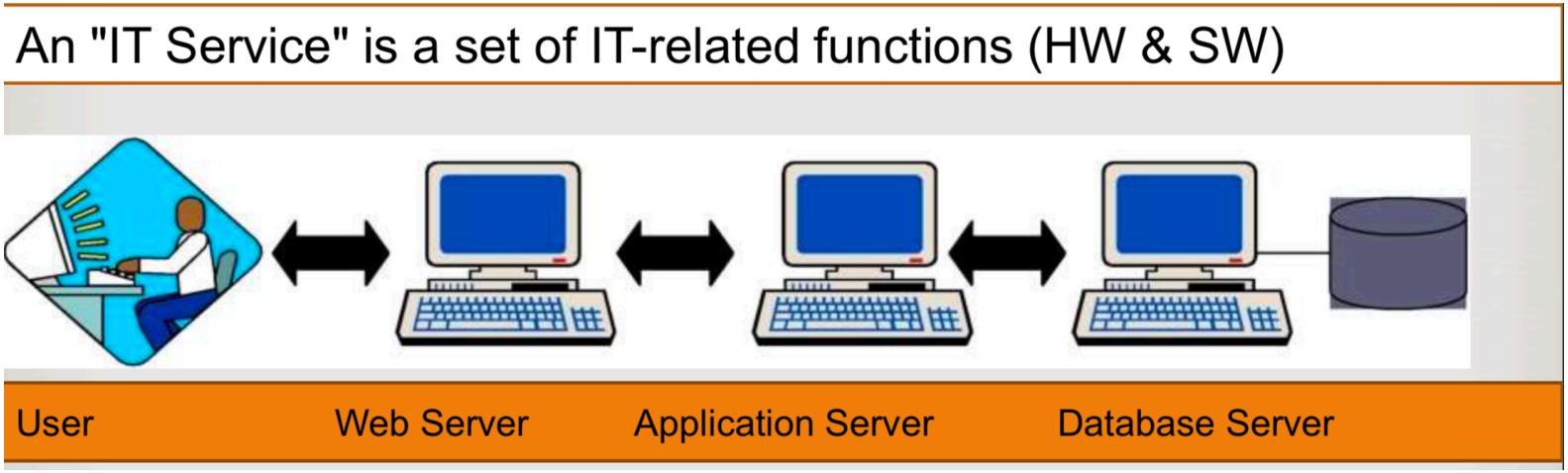
Kakao workshop, 2014

Terms

- Programable Resource Management
  - Something for the computing (calculate, store)
  - Something Measurable
    - # of VM or Container can be resource
  - Something Atomic (eg. IP address, block)
    - CPU can’t be resource from VM’s side of view, CPU could be resource from container’s

Second, of All

- We call Cloud ‘ … as a service’,
- What is service ?



- Now, you know what is micro-service

if (kakao) : Cloud.define()

Programmable **Resource** Management	
2015	What is Resource?

Programmable Resource **Life Cycle** Management	
2017	What is Life Cycle?

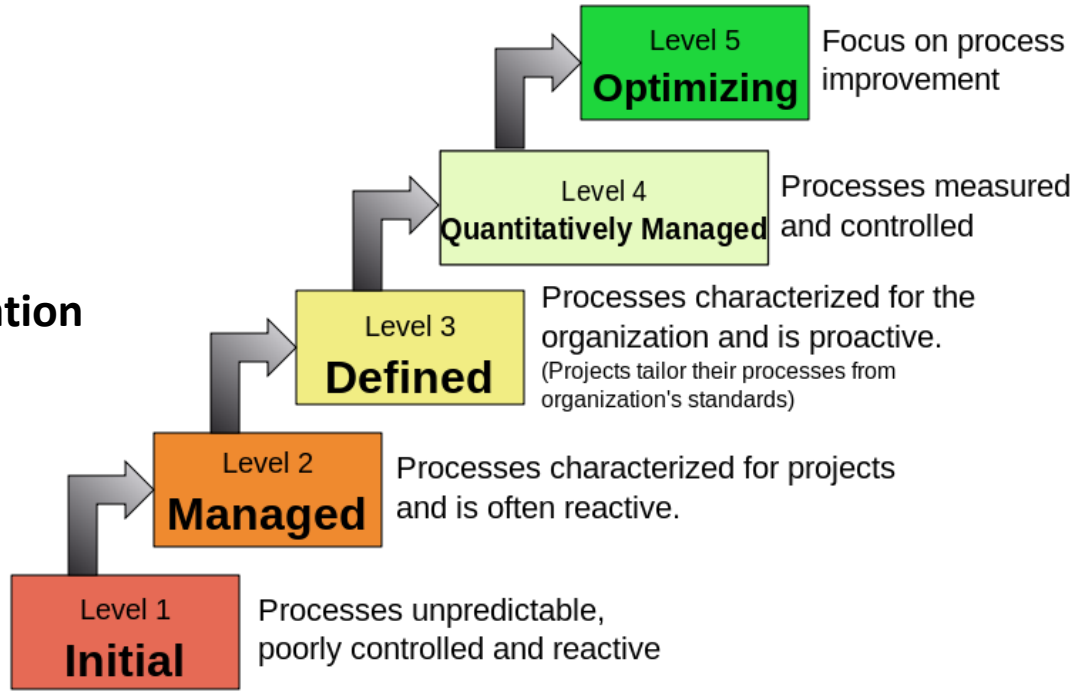
Programmable **Service** Management	
2018	What is Service?

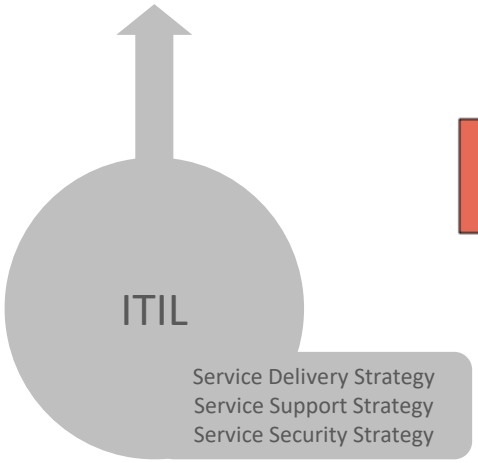
How Far can you go with your cloud?

CMMI Model

Capability Maturity Model Integration

developed by CMU

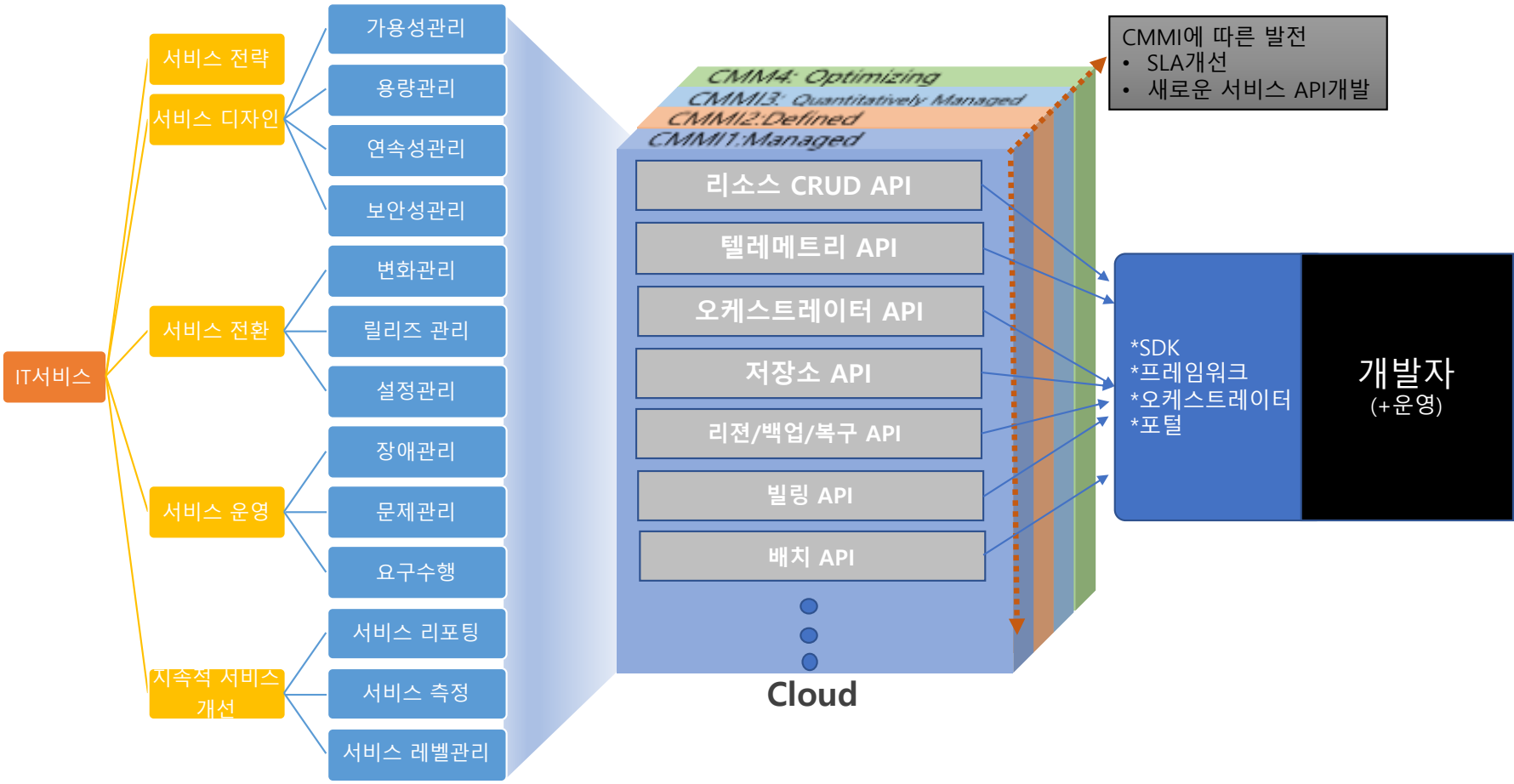




Now it’s time to ask

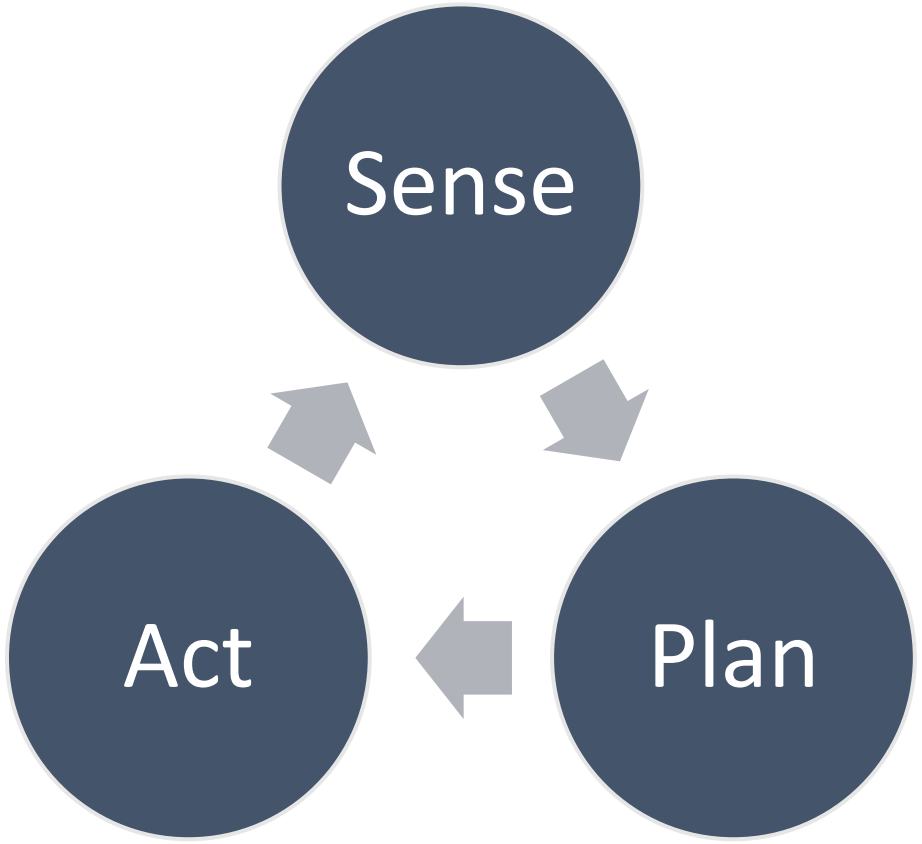
What’s for?

The overall IT strategy with cloud

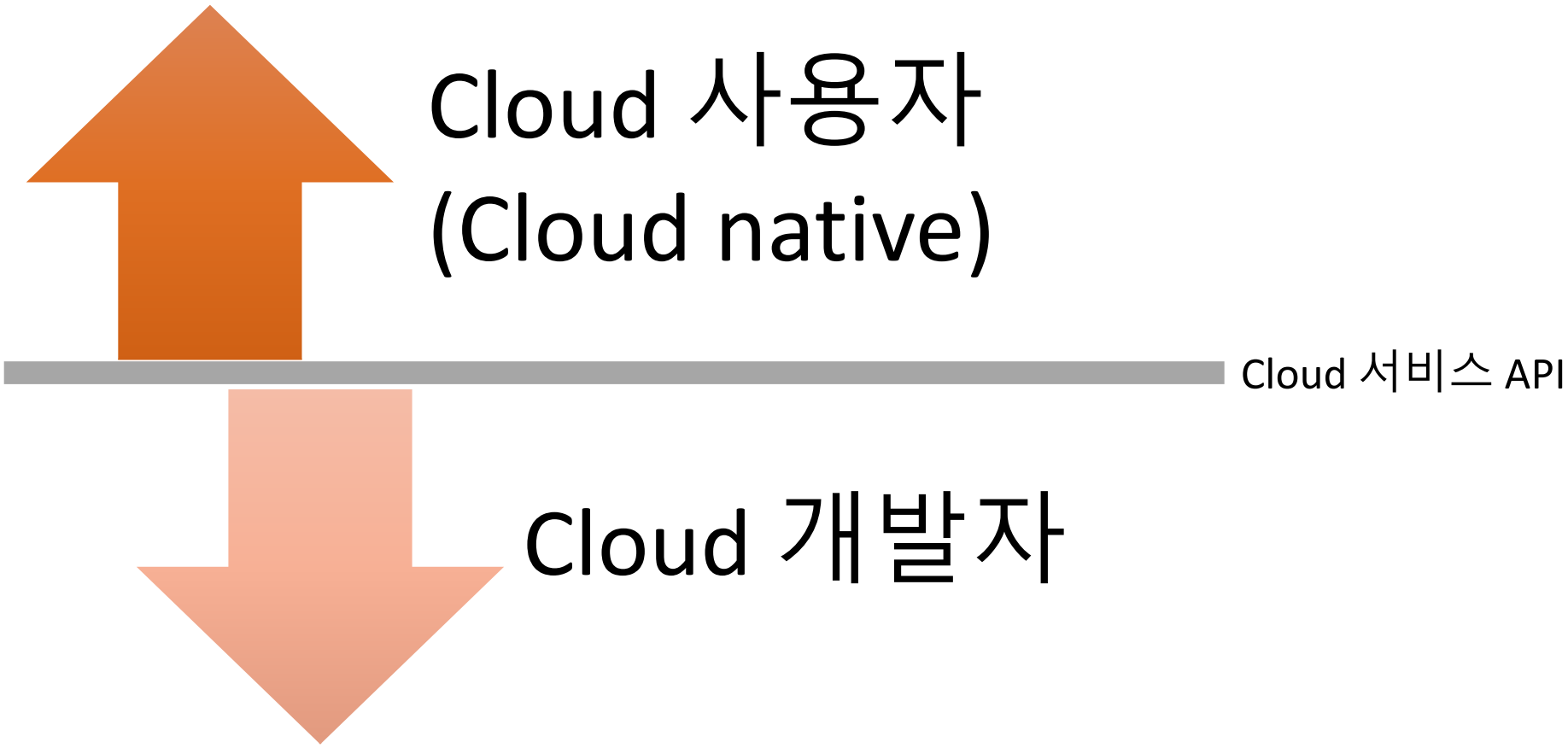


The base of ITIL/CMMI is:

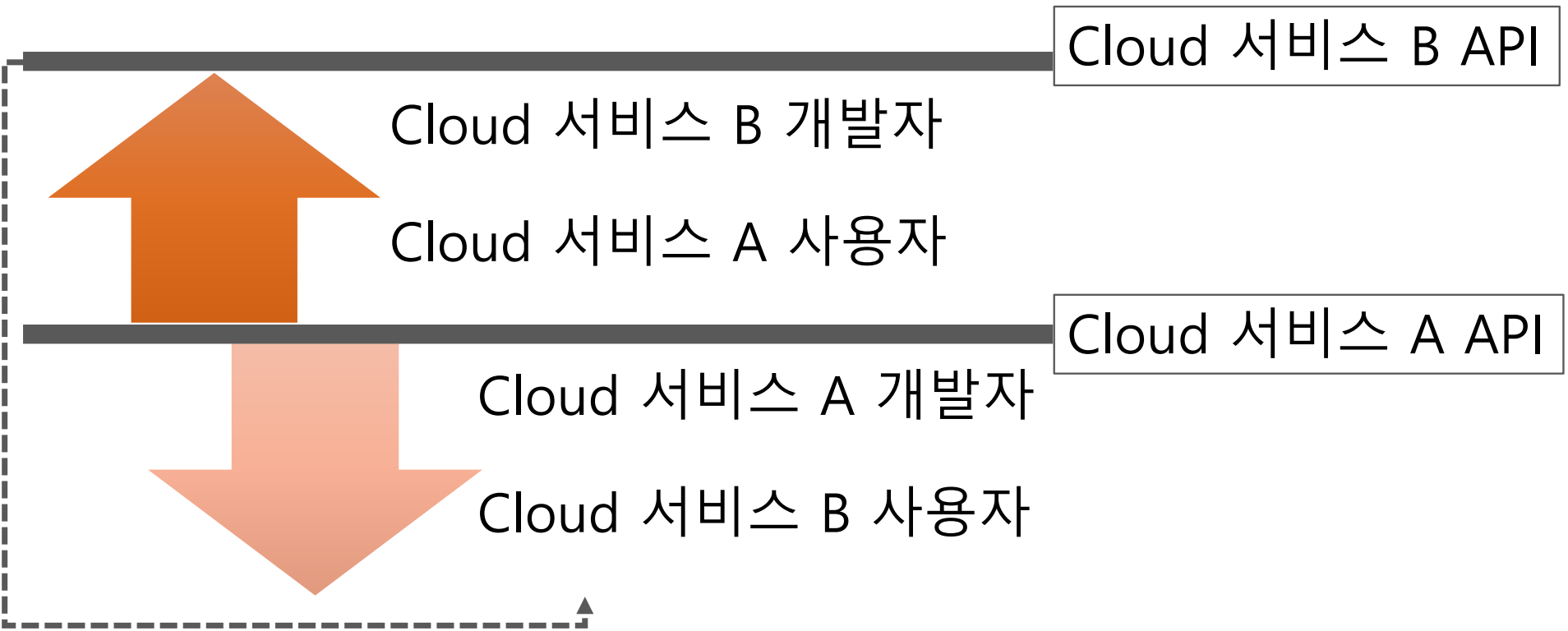
Continual Improvement



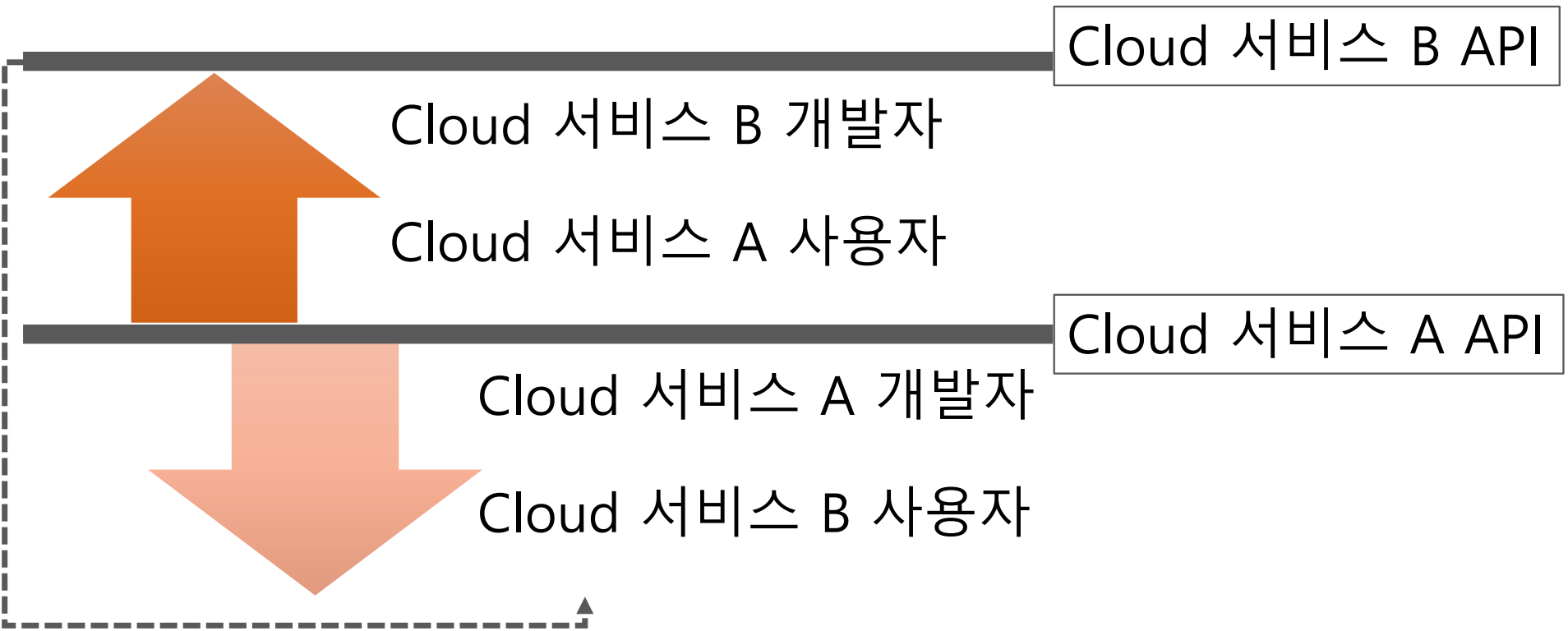
Cloud vs Cloud Native



Cloud vs Cloud Native, The Reality



Cloud vs Cloud Native, The Reality



CMMI Dev perspective

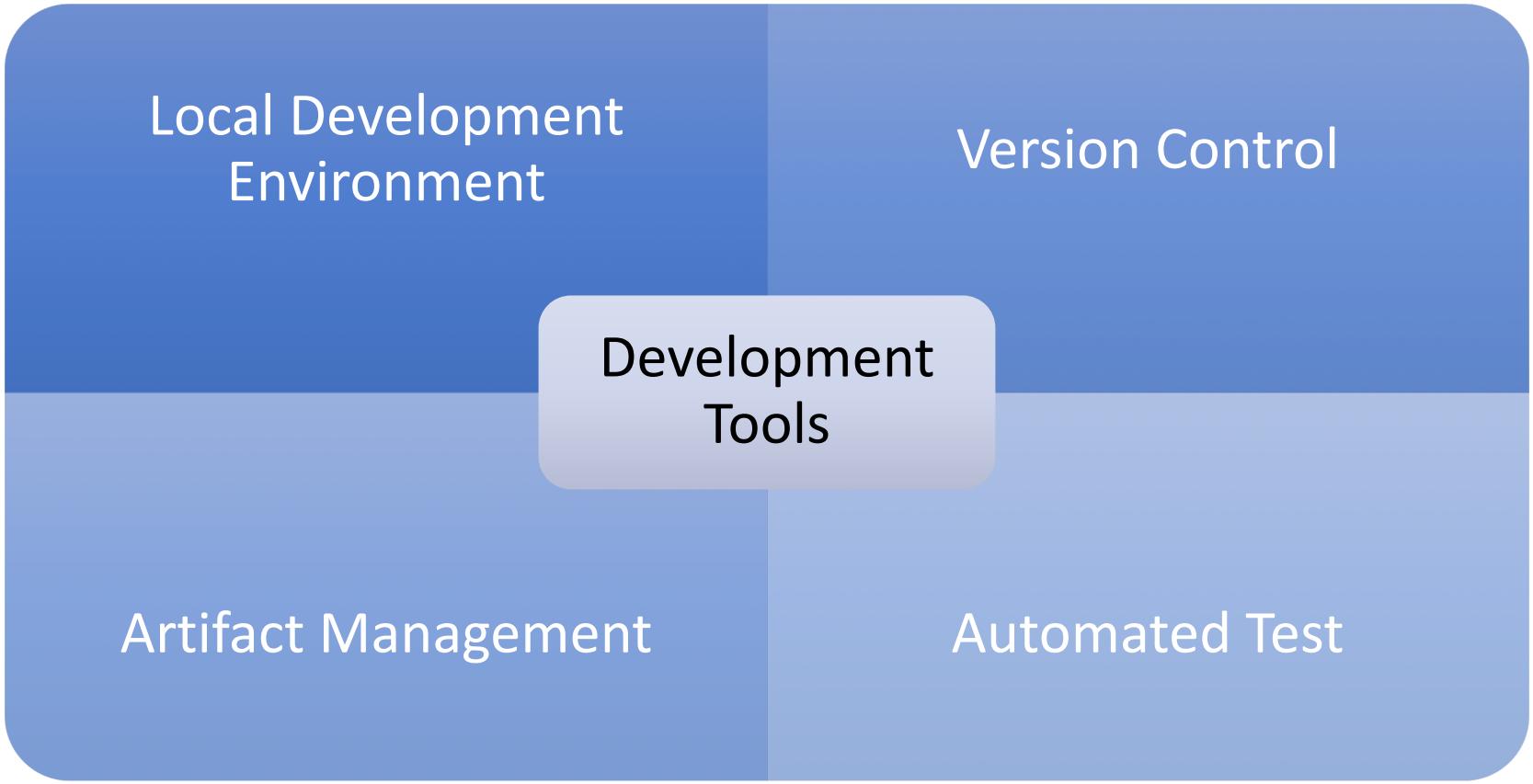
Maturity → Targets

	Initial	Managed	Defined	Quantitatively Managed	Optimizing
Culture & Organization	• Teams organized based on platform/ technology • Defined and documented processes	• One backlog per team • Adopt agile methodologies • Remove team boundaries	• Extended team collaboration • Remove boundary dev/ ops • Common process for all changes	• Cross-team continuous improvement • Teams responsible all the way to production	• Cross functional teams
Build & Deploy	• Centralized version control • Automated build scripts • No management of artifacts • Manual deployment • Environments are manually provisioned	• Polling CI builds • Any build can be re-created from source control • Management of build artifacts • Automated deployment scripts • Automated provisioning of environments	• Commit hook CI builds • Build fails if quality is not met (code analysis, performance, etc.) • Push button deployment and release of any releasable artifact to any environment • Standard deployment process for all environments	• Team priorities keeping codebase deployable over doing new work • Builds are not left broken • Orchestrated deployments • Blue Green Deployments	• Zero touch Continuous Deployments
Release	• Infrequent and unreliable releases • Manual process	• Painful infrequent but reliable releases	• Infrequent but fully automated and reliable releases in any environment	• Frequent fully automated releases • Deployment disconnected from release • Canary releases	• No rollbacks, always roll forward
Data Management	• Data migrations are performed manually, no scripts	• Data migrations using versioned scripts, performed manually	• Automated and versioned changes to datastores	• Changes to datastores automatically performed as part of the deployment process	• Automatic datastore changes and rollbacks tested with every deployment
Test & Verification	• Automated unit tests • Separate test environment	• Automatic Integration Tests • Static code analysis • Test coverage analysis	• Automatic functional tests • Manual performance/ security tests	• Fully automatic acceptance tests • Automatic performance/security tests • Manual exploratory testing based on risk based testing analysis	• Verify expected business value • Defects found and fixed immediately (roll forward)
Information & Reporting	• Baseline process metrics • Manual reporting • Visible to report runner	• Measure the process • Automatic reporting • Visible to team	• Automatic generation of release notes • Pipeline traceability • Reporting history • Visible to cross-silo	• Report trend analysis • Real time graphs on deployment pipeline metrics	• Dynamic self-service of information • Customizable dashboards • Cross-reference across organizational boundaries

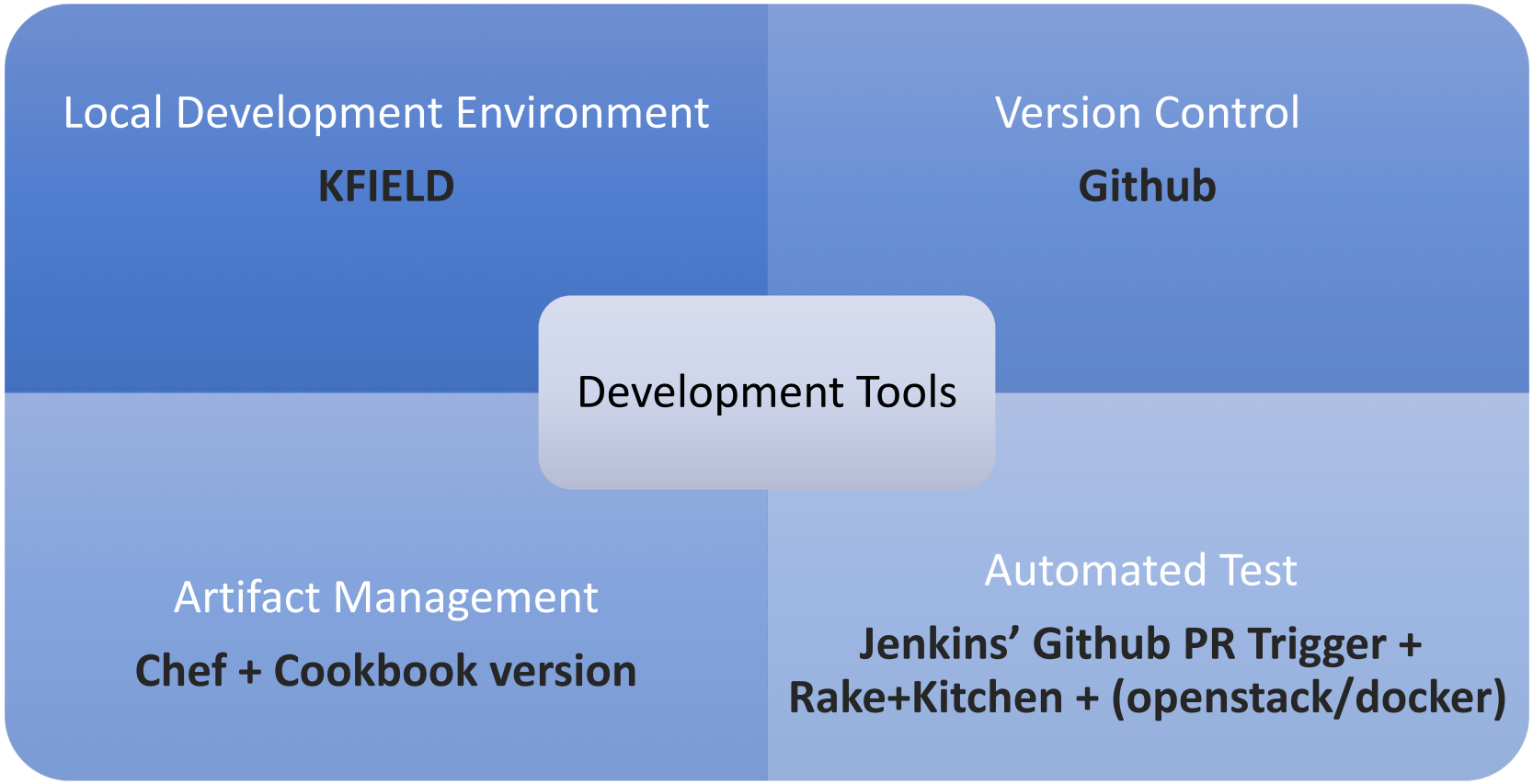
To do it faster!

- Time machine will be needed

Tools for time machine



Tools: kakao cloud

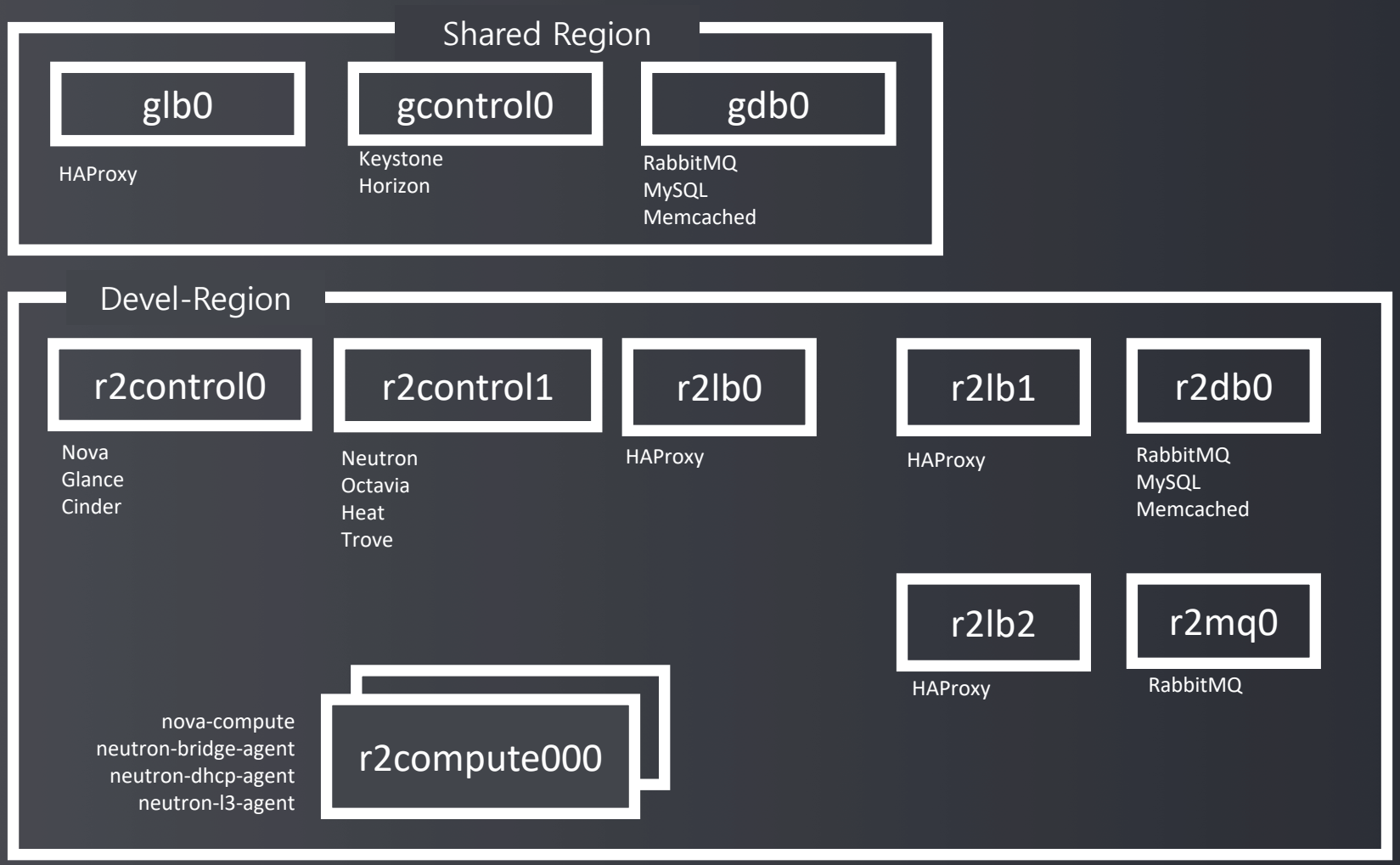


Kfield, Cloud Native framework

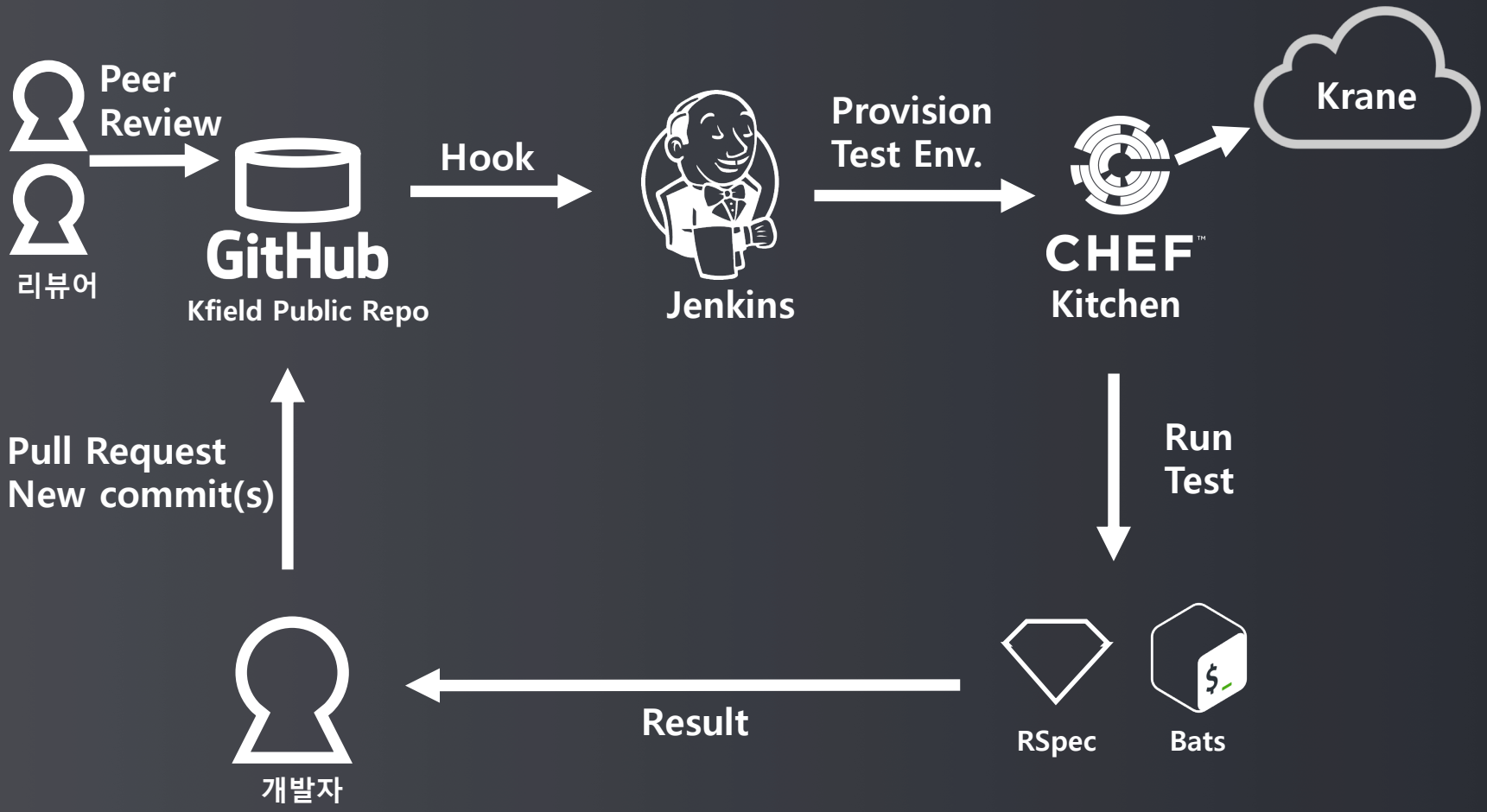
Rails

Tox

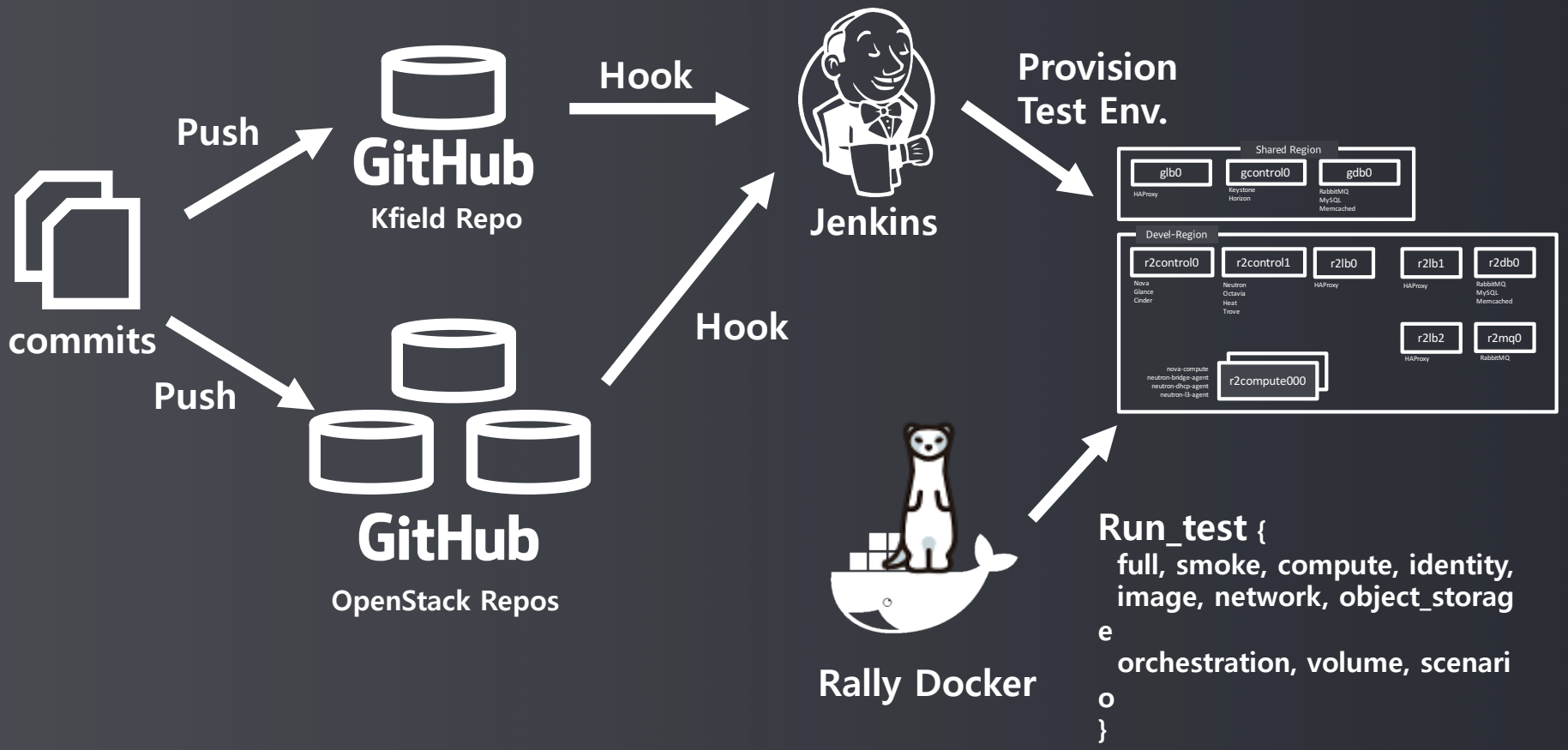
KField, Cloud Native framework



Kfield, Integration Test Pipeline



KField, More Integration Test

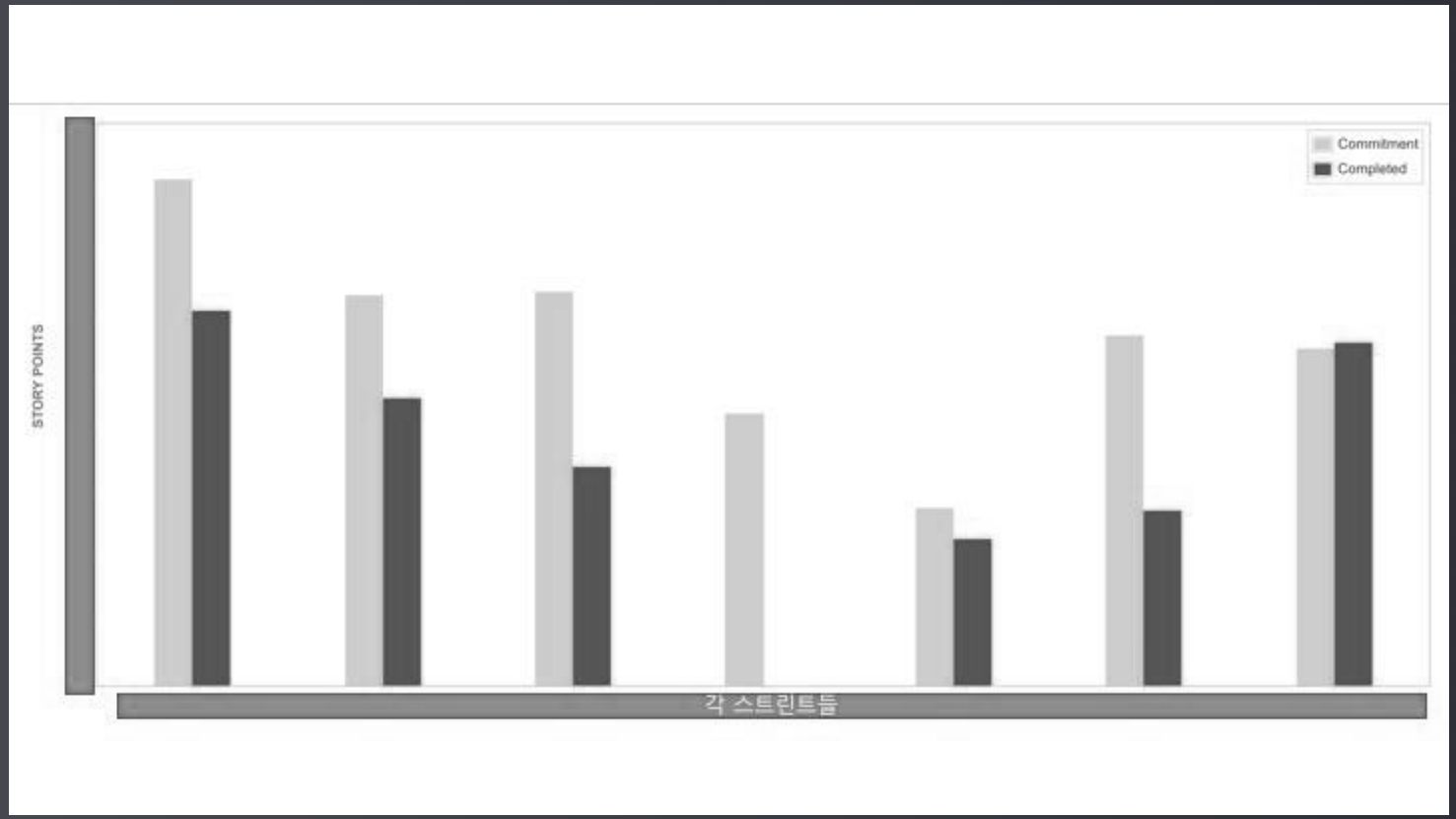


Did you consider agile dev. process?

- Scrum
- Daily Standup
- Sprint
- Ticket and Story point
- But, Why?
- Job evaluation accuracy based on ticket count, aggregate story point
  - 88%
  - Anyway, Dev performance should be measured

Why we need all these?

- Service is Permanent, Human is ephemeral

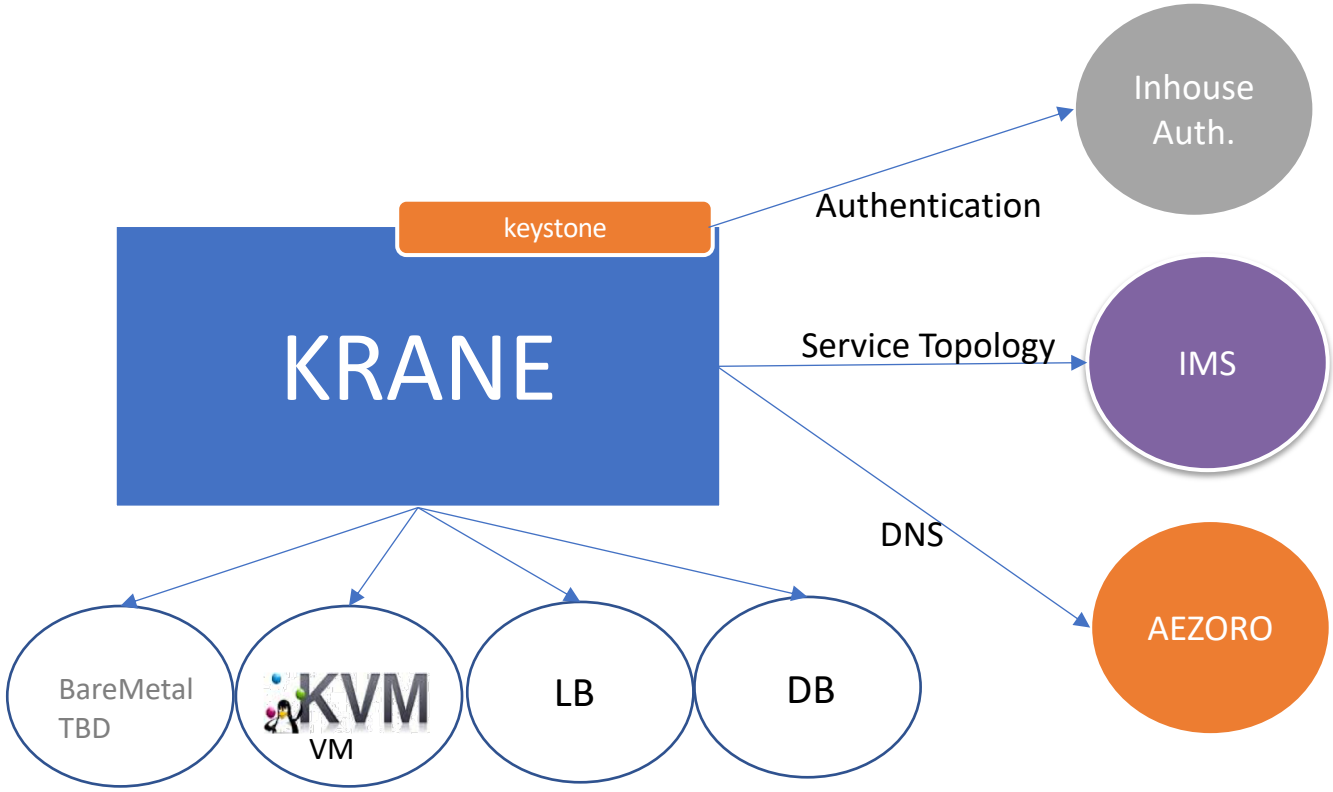


if (kakao): IaaS

What is the purpose of doing IaaS?

CMM0	CMM1
legacy	self service Dev resource
output: ITF	output: krane (openstack cloud)

if (kakao): IaaS (service name KRANE)

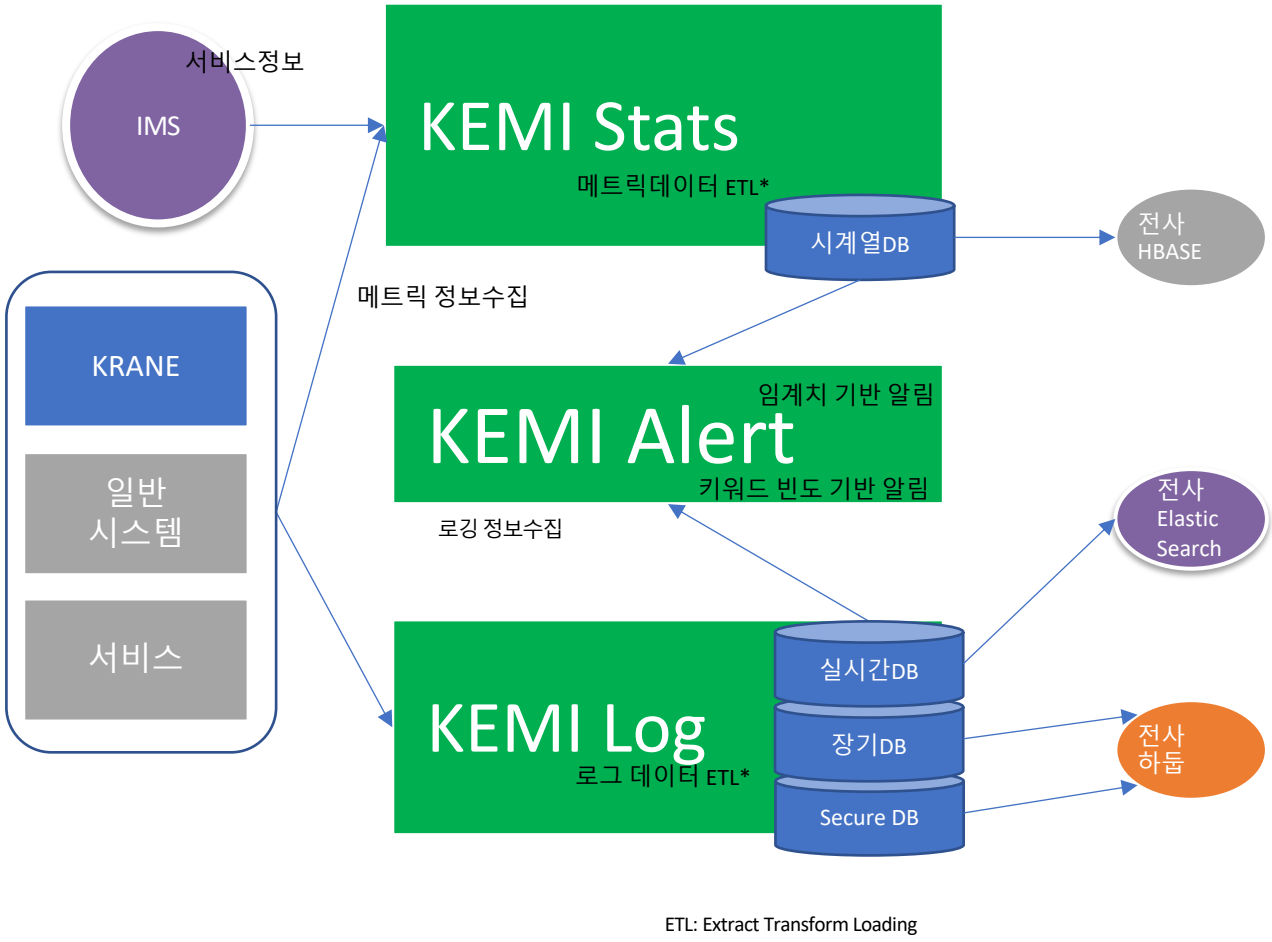


Cloud CMMI2-MaaS

What is the sole purpose of doing MaaS?

CMM0	CMM1	CMM2
legacy	self service Dev resource	limited Prod resources
output: cloudTF	output: krane (openstack cloud)	output: kemi (MaaS)

if (kakao): MaaS (service name KEMI)



ETL: Extract Transform Loading

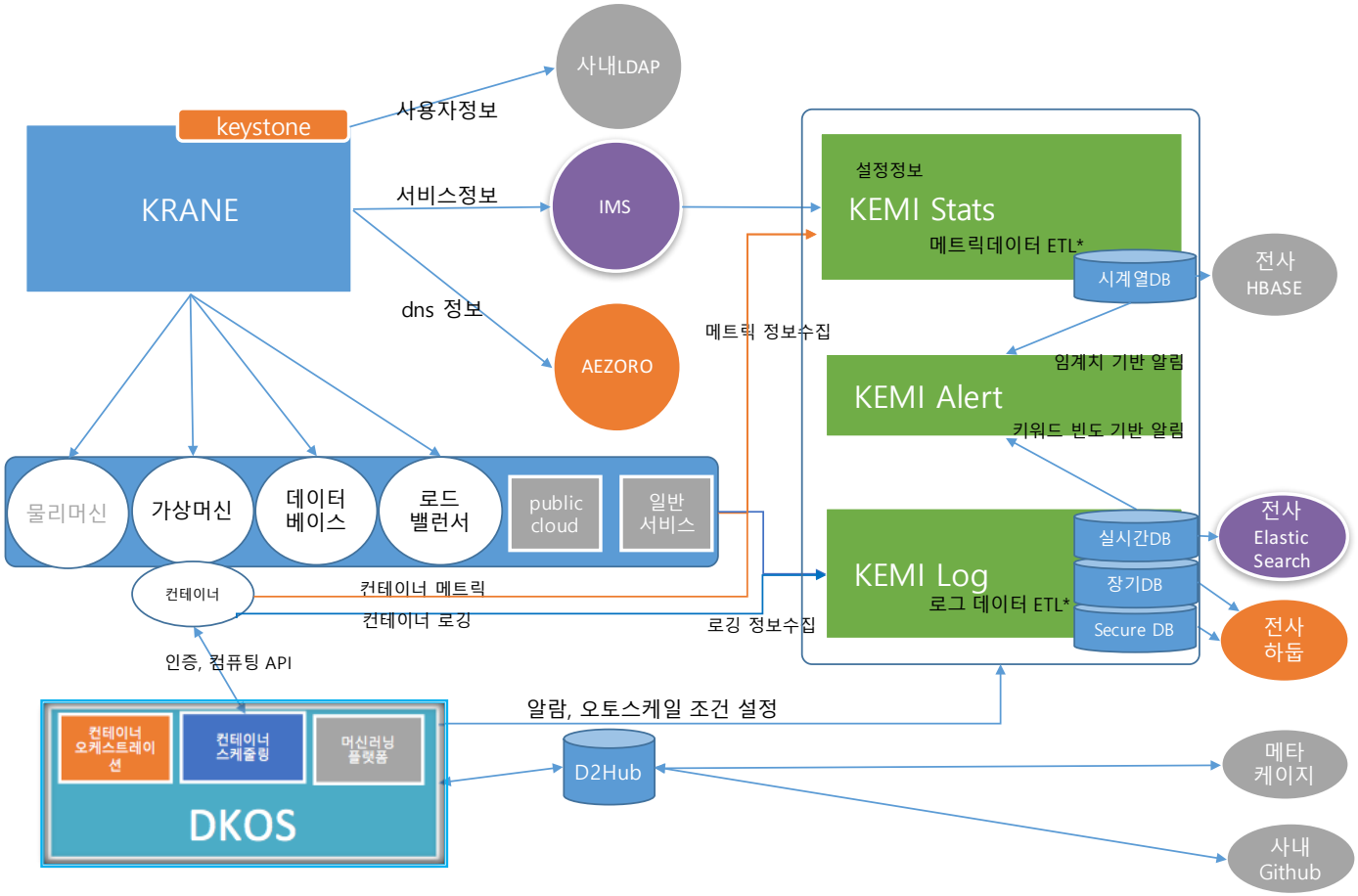
Cloud CMMI3-CaaS

What is the sole purpose of doing CaaS(Container As A Service)?

CMM0	CMM1	CMM2	CMM3
legacy	self service Dev resource	limited Prod resources	Automated CloudUsage
output: cloudTF	output: krane (openstack cloud)	output: kemi (MaaS)	output: DKOS (CaaS)

if (kakao): DKOS(The key.)

DKOS is connected to everything



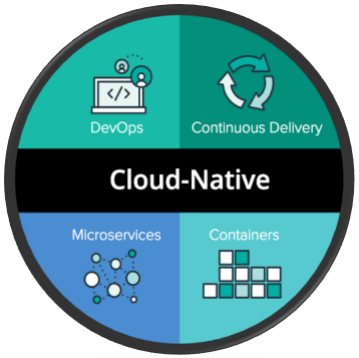
Cloud CMMI4-Cloud Native Platform

What is the sole purpose of doing Cloud Native Platform?

CMM0	CMM1	CMM2	CMM3	CMM4
legacy	self service Dev resource	limited Prod resources	Automated CloudUsage	Integrated Service Platform
output: cloudTF	output: krane (openstack cloud)	output: kemi (MaaS)	output: DKOS (CaaS)	output: 9rum (C.N.P)

Little bit more specific

- Cloud Native Applications Characteristics
  - Micro-services
  - Health Reporting
  - Telemetry Data
  - Resiliency
  - Declarative



https://pivotal.io/cloud-native)

↳ First Ask ‘Why we need all these?’

↳ Dynamically Orchestrated Platform

if (kakao):
Characteristics to Requirement

- C. N. Applications Characteristics
  - Microservices
  - Health Reporting
  - Telemetry Data
  - Resiliency
  - Declarative

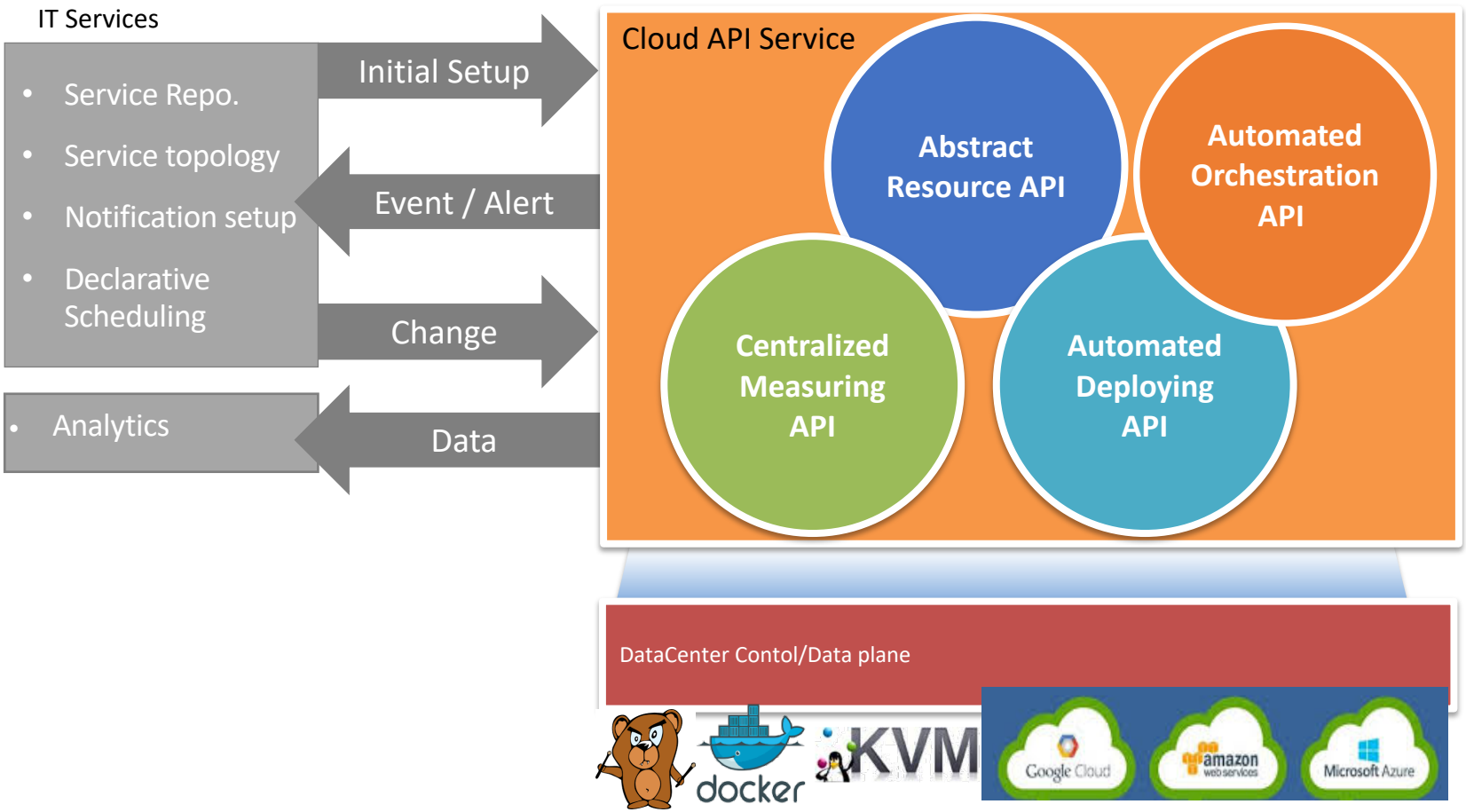
- Platform Requirement
  - Isolation
  - Resource Allocation /Scheduling
  - Service Discovery
  - Monitoring/Logging
  - Metric Aggregation
  - Debugging and Tracing

Cloud CMMI4-Cloud Native Platform

What is the sole purpose of doing Cloud Native Platform?

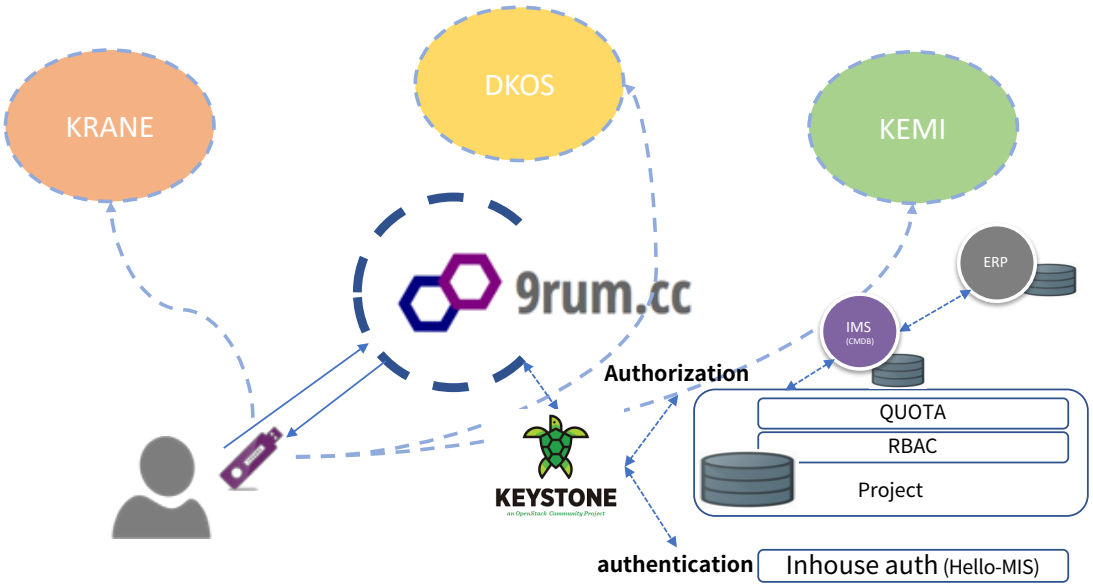
CMM0	CMM1	CMM2	CMM3	CMM4
legacy	self service Dev resource	limited Prod resources	Automated CloudUsage	Integrated Service Platform
output: cloudTF	output: krane (openstack cloud)	output: kemi (MaaS)	output: DKOS (CaaS)	output: 9rum (C.N.P)

Cloud is heading for
Dev can Ops over hybrid environment



Key technology for Hybrid cloud?

- Pretty sure, Network is not
- Integrated Authentication (so called SSO), Integrated Measuring



Some outputs:

Cloud API coverage (%)

Container slave ratio over Cloud (%)

Indexing Data (TB/Day)

Expense $Million/Yr

(compared to public cloud, excluding traffic/storage)

License Expense

QnA