Fundamentals of Cloud Computing, and Manufacturing Services Automated Construction Scheme (MSACS)

Yu-Chuan Lin

2020/11/12

Outline

I. Industry 4.1

II. Introduction to Internet of Things

III. Introduction to Edge Computing

IV. Introduction to Cloud Computing

V. Cloud-based AVM System

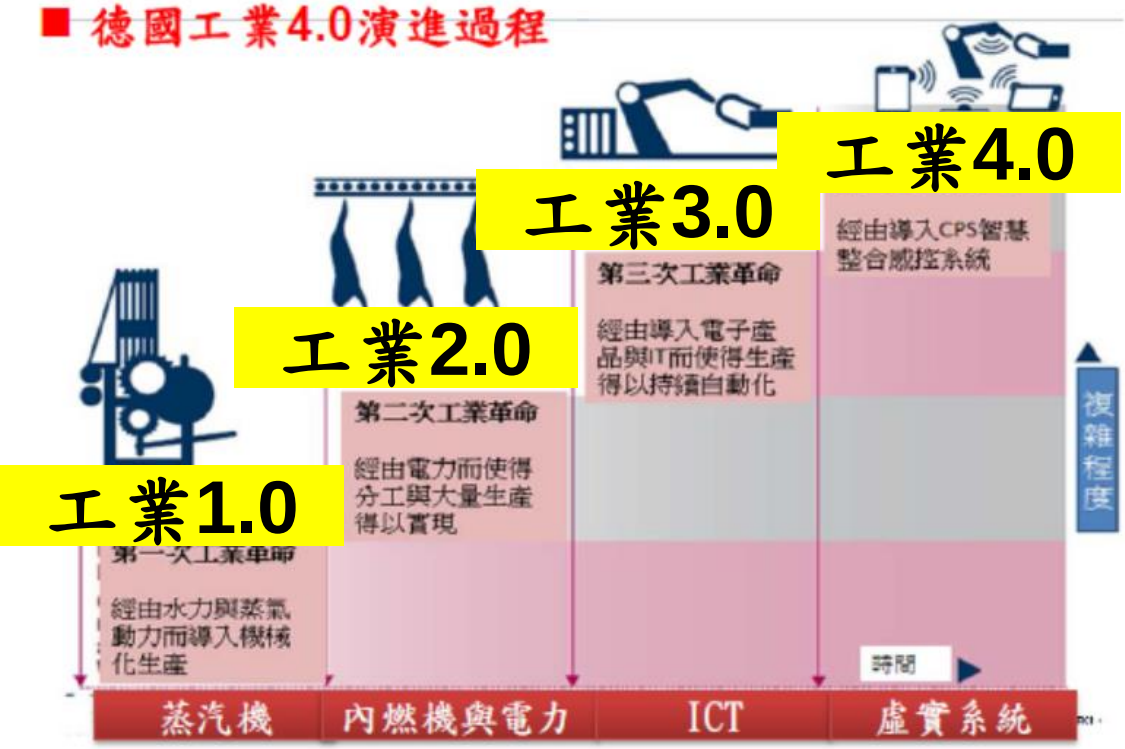
VI. MSACS (Manufacturing Services Automated Construction Scheme)

Industry 4.0

(第4次工業革命)

德國工業4.0-演進過程

■ 德國工業4.0演進過程



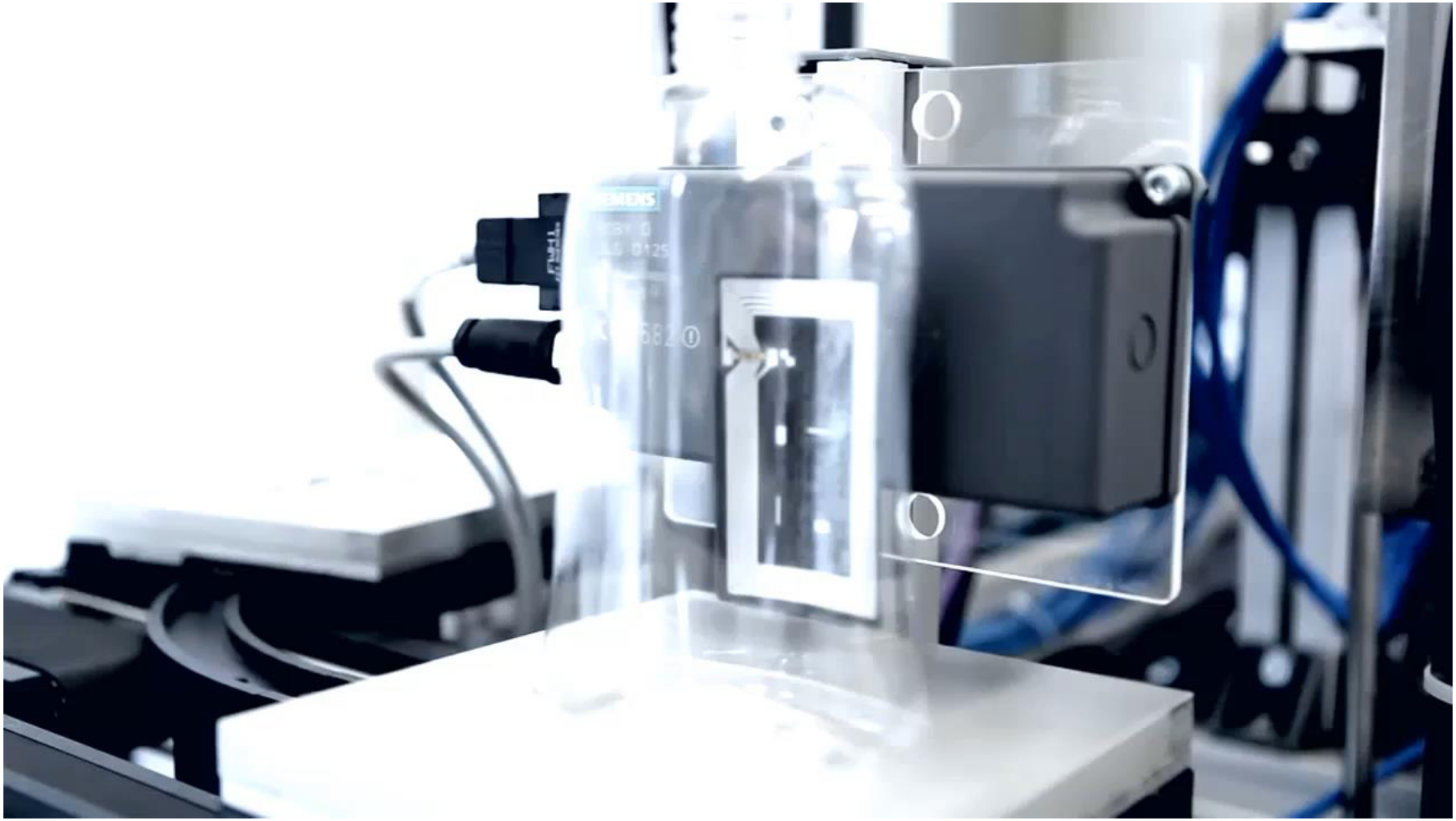
- 「工業4.0」為德國聯邦政府在2012年時提出落實「2020高科技戰略」的十大未來計畫之一。
- 在生產製造過程中，整合計算、通訊與控制的虛實化系統(CPS)，連結物聯網，建構「智慧工廠」，形成智慧製造與服務的全新商機與商業模式。

Industry 3.0	Industry 4.0
■ 中央控制	network達成分散自我控制
■ 價值創造鏈	虛擬的Organizations
■ 預先規劃運作生產系統	生產單位具自我組織的自主能力
■ 產品為加工的被動物件	產品決定智慧生產過程
■ 僵化員工現場工作方式	人機協同彈性作業

Source: 經濟部工業局，”從Industry 4.0看台灣生產力推升的契機"，2014年9月19日。

什麼是工業4.0

工業4.0之智慧工廠運作情境影片

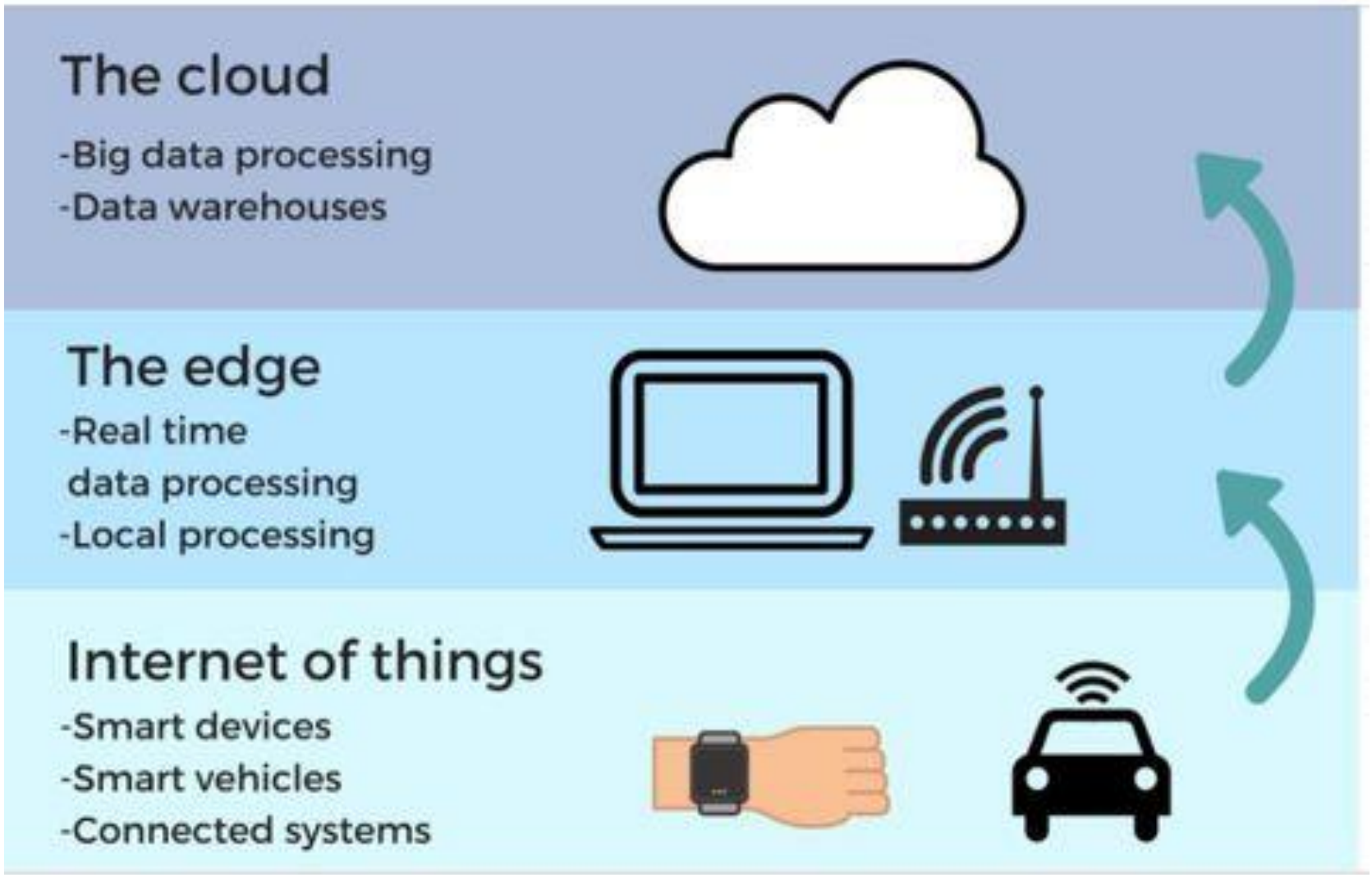


Source: Siemens, URL: http://www.siemens.com/

工業4.0 (Industry 4.0)

- 德國於2012年推動【工業4.0計畫】(Industry 4.0)，為其落實【2020高科技戰略】的十大未來計畫之一。
- 工業4.0：
  - 以物聯網等感測技術技術連結萬物，更可透過雲端運算、大數據與人工智慧技術，讓機械與機械、機械與人之間可以相互溝通，進行自主管理與改善，建構成一個智慧工廠，將傳統生產方式轉為高度客製化、智慧化、服務化的商業模式，可以快速製造少量多項的產品，因應快速變化的市場。

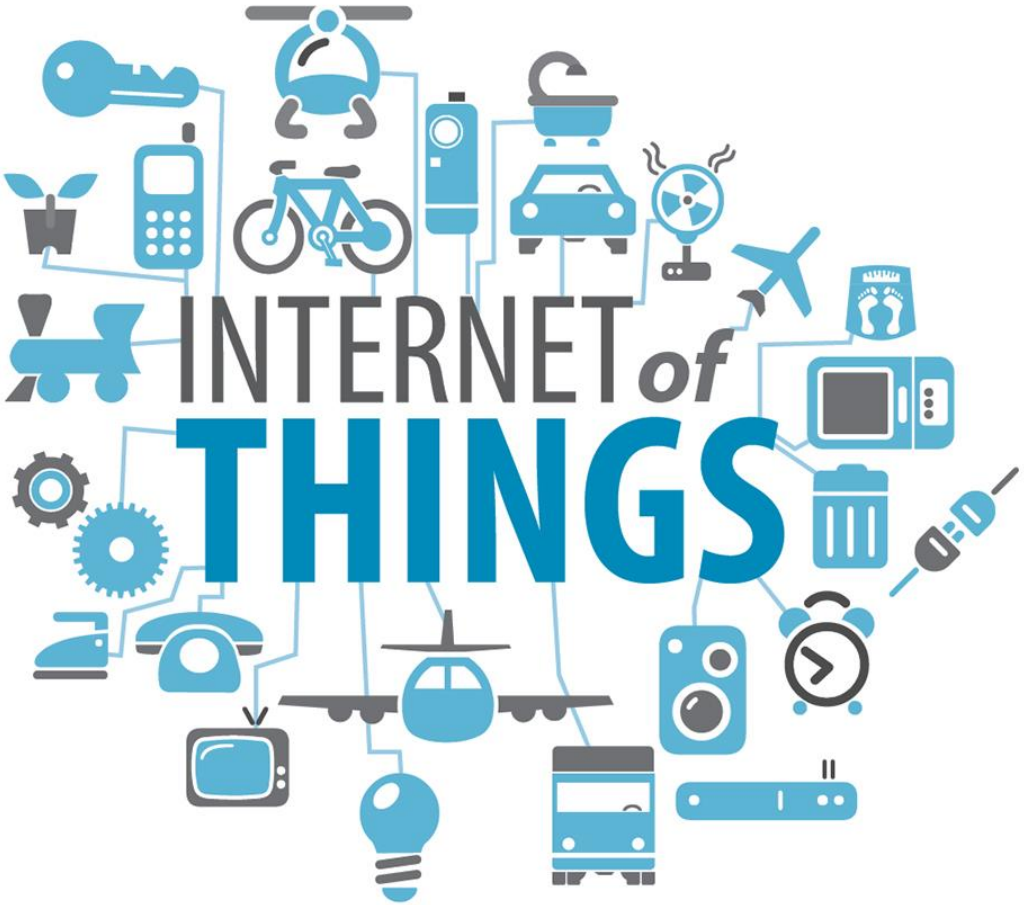
工業4.0內之三大要素：
物聯網、邊緣運算與雲端運算



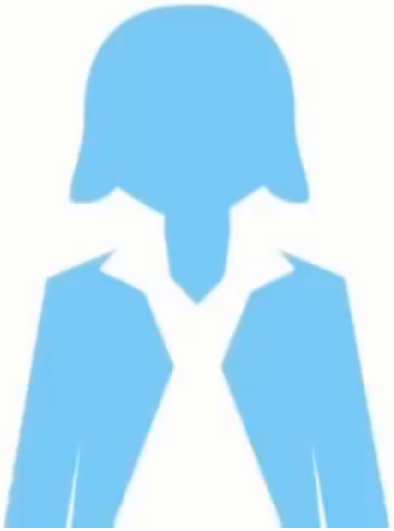
物聯網
(Internet of Things)

物聯網應用

物聯網的應用範圍包括：智慧製造、交通、醫療、智慧電網、智慧生活、智慧建築、農牧業、金融與服務業、國防軍事等應用領域。



物聯網應用 - 汽車業、運輸業、零售業

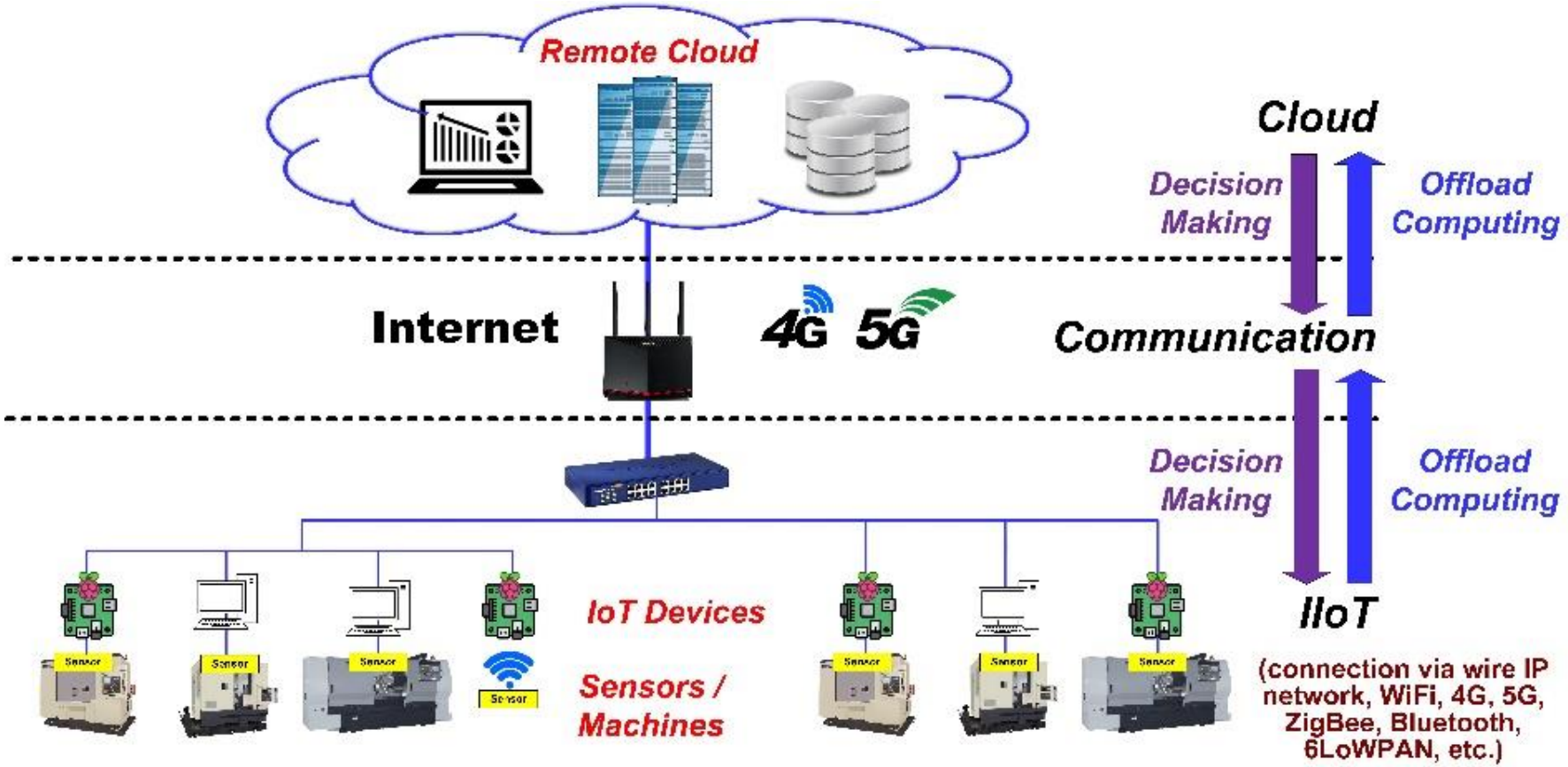


IBM物聯網應用 - 汽車業、運輸業、零售業 https://www.youtube.com/watch?v=6KSiKQxTXQQ

什麼是物聯網？

- **1998年，美國麻省理工學院Auto-ID中心主任愛斯頓（Kevin Ashton）提出物聯網（Internet of Things，簡稱IoT）一詞。**
- **顧名思義：讓萬物之間利用網路全部結合在一起**
- 在物聯網下，可以賦予家電、裝置、設備生命。回家前可事先用手機開啟冷氣，空氣品質不好，會自動打開空氣清淨機，久坐沒起身運動，手錶會通知你起來走走，任何你可以想到的運用以及物品，都可以結合。**物聯網帶來更方便的生活**

物聯網 的應用


Remote Cloud
Cloud
Decision Making
Offload Computing
Internet
4G
5G
Communication
Decision Making
Offload Computing
IIoT
IoT Devices
Sensors / Machines
(connection via wire IP network, WiFi, 4G, 5G, ZigBee, Bluetooth, 6LoWPAN, etc.)

邊緣運算
(Edge Computing)

邊緣運算（概念篇）

DIGITIMES
物聯網

"邊緣運算
Edge Computing"

實踐工業4.0不可不知

邊緣運算（工業現場篇）

DIGITIMES
物聯網

"邊緣運算
Edge Computing"

實踐工業4.0不可不知

邊緣運算（應用篇，智慧電梯）

DIGITIMES
物聯網

"邊緣運算
Edge Computing"

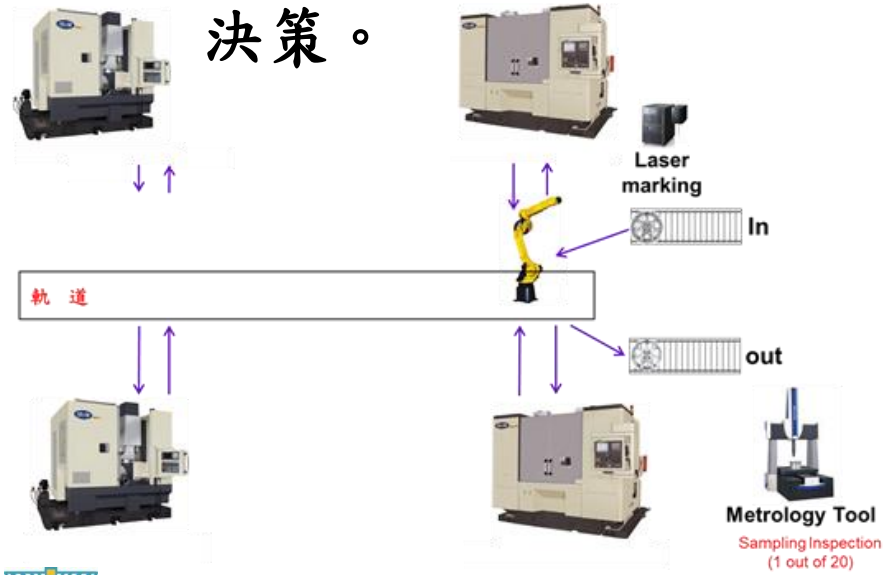
實踐工業4.0不可不知

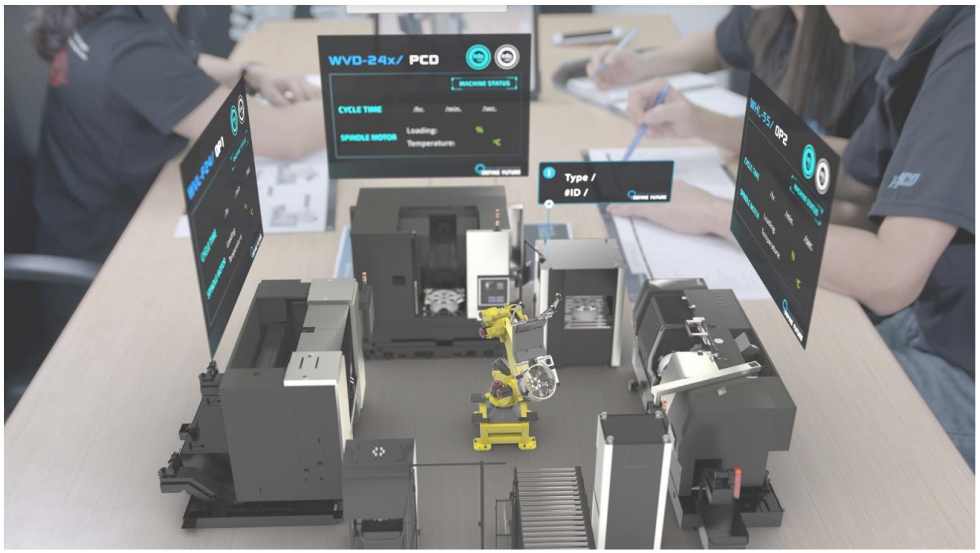
ThyssenKrupp

- New vision to elevator maintenance with Microsoft HoloLens

機台遠距管理和維護

- 隨著國際化的趨勢，工具機廠商將其產品銷售至國外；且必須具備管理散佈各地之多廠區生產線時，如何執行機台遠距管理和維護，就變得相當重要。
- 透過AR/VR及資通技術整合，呈現完整的自動化生產資訊管理模組，如即時狀態呈現、製造流程可視化...等，便於客戶應用於管理、維修、和保養，讓各領域管理人員可透過即時具體影像資訊進行判斷與溝通，作出精準決策。

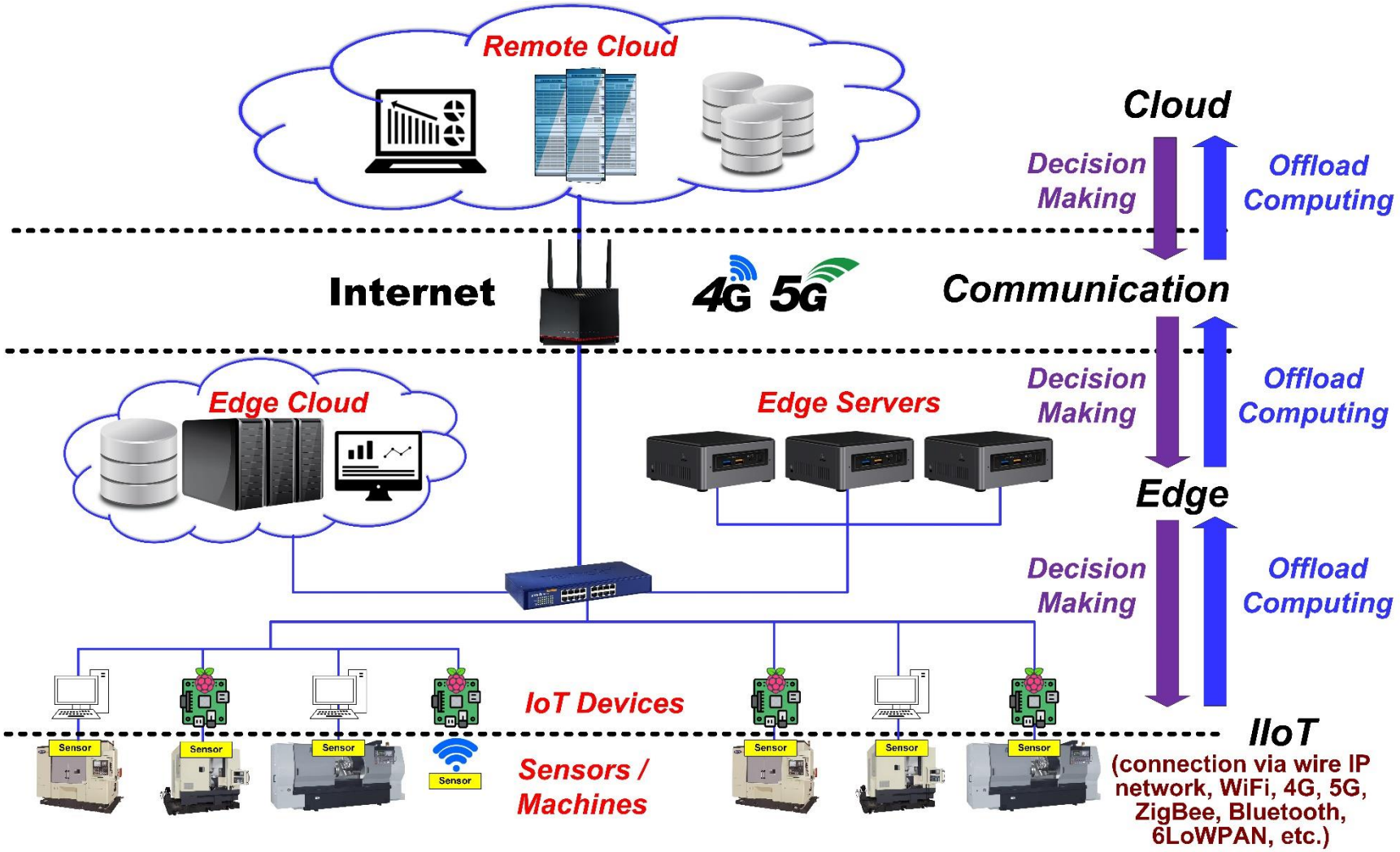




邊緣運算(Edge computing)定義

- 邊緣運算(Edge computing)，是一種分散式運算的架構，將應用程式、數據資料與服務的運算，由網路中心節點，移往網路邏輯上的邊緣節點來處理。(維基百科)
- 有了 Edge 端直接針對多裝置、龐大訊息先做擷取、過濾與處理，對裝置端做出回饋與反應，不用讓所有資料都上到雲端，以期在資料量逐漸龐大、重視資訊即時處理傳輸的現代，更能有效率處理資訊，減少事事上雲端所帶來的時間遞延與資料傳輸/儲存成本。

邊緣運算的應用


Remote Cloud
Cloud
Decision Making
Offload Computing
Internet
4G 5G
Communication
Edge Cloud
Edge Servers
Decision Making
Offload Computing
Edge
Decision Making
Offload Computing
IoT Devices
Sensors / Machines
Sensor
IIoT
(connection via wire IP network, WiFi, 4G, 5G, ZigBee, Bluetooth, 6LoWPAN, etc.)

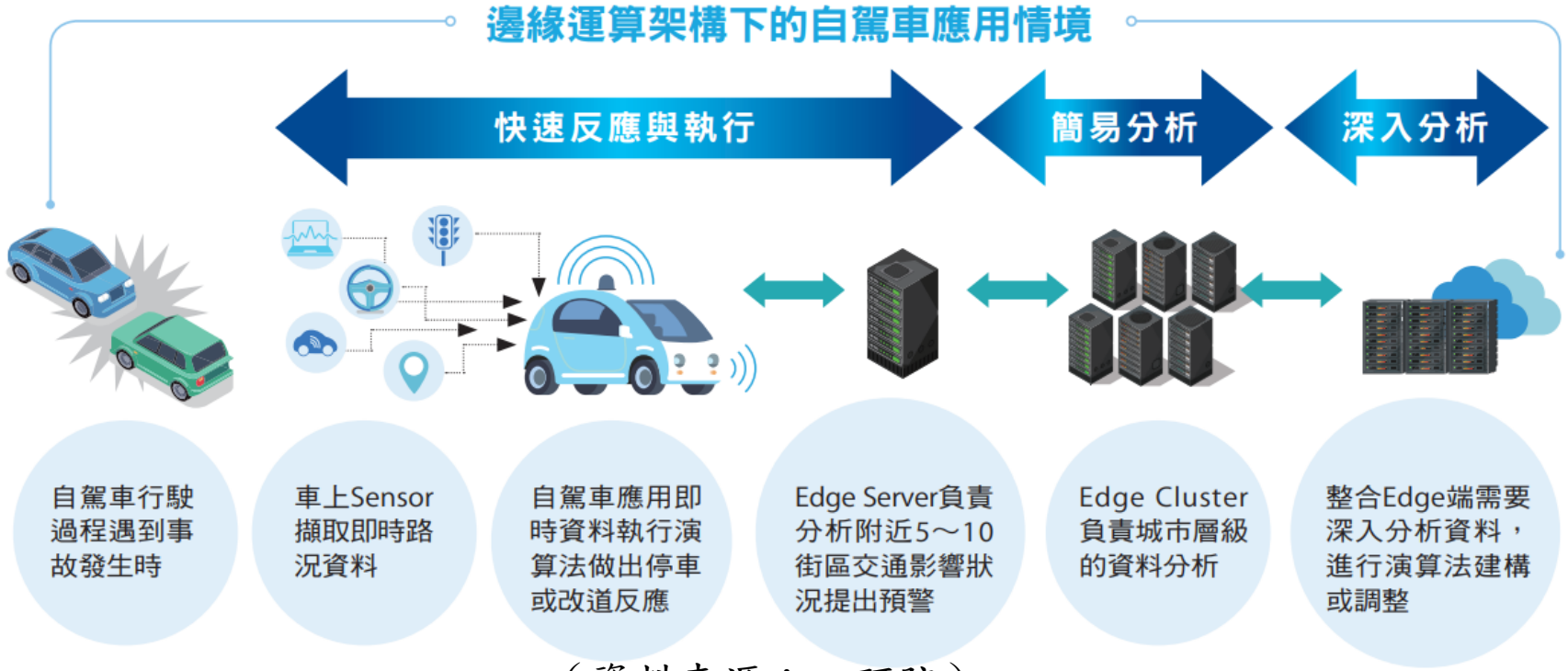
AWS 邊緣運算服務 Wavelength

- AWS 邊緣運算服務 Wavelength 於美正式開放
  - 2020 年 8 月雲端服務商 Amazon Web Services（AWS）宣布邊緣運算服務 Wavelength 正式於波士頓和舊金山區啟動，將 AWS 運算與儲存服務部署至電信營運商的 5G 網路邊緣與資料中心，開發人員可利用 AWS API 和開發工具，直接於 5G 網路建置應用程式，因此應用程式流量可以從 5G 裝置傳送至 Wavelength 區域中執行的應用程式伺服器，無須離開電信網路。這避免了因應用程式流量必須周遊網際網路多次跳轉才能到達目的地而導致的延遲。

（資料來源：拓墣產業研究院）

邊緣運算的應用-自駕車

- 以自駕車為例，接近用戶端的終端(如感測器)以及聯物網裝置(IoT)、邊緣伺服器(Edge Server)等，負責擷取路況和鄰近街區資料並執行演算法，以做出快速判斷和反應；更上層的邊緣電腦叢集(Edge Cluster)則是負責城市等較大範圍資料；而更深入的分析或是演算法的學習調整，就交給最上層的雲端來完成。



（資料來源：工研院）

資料來源：Gartner；工研院IEK

雲端運算
(Cloud Computing)

雲端運算與其應用

- 雲端運算(Cloud Computing)是目前網際網路運用之主流方式。
- 其概念為將運算資源與資料從本地端電腦移轉到大型的資料中心，並透過虛擬化軟體技術與網際網路提供多樣化的服務，為個人和企業的使用者提供可隨需(On-demand)存取的運算與儲存資源。
- 雲端運算已應用於各個領域(含個人使用)。

雲端運算的定義

雲端運算定義 (by NIST)：

- 雲端運算為一種資源共享模式，能夠方便地隨需利用網路存取共享的廣大運算資源（如網絡、伺服器、儲存、應用程式及服務等）。
- 可透過最少的管理工作來服務供應者之互動需求，以便能快速地提供各項資訊服務。

雲端運算的特性與優點

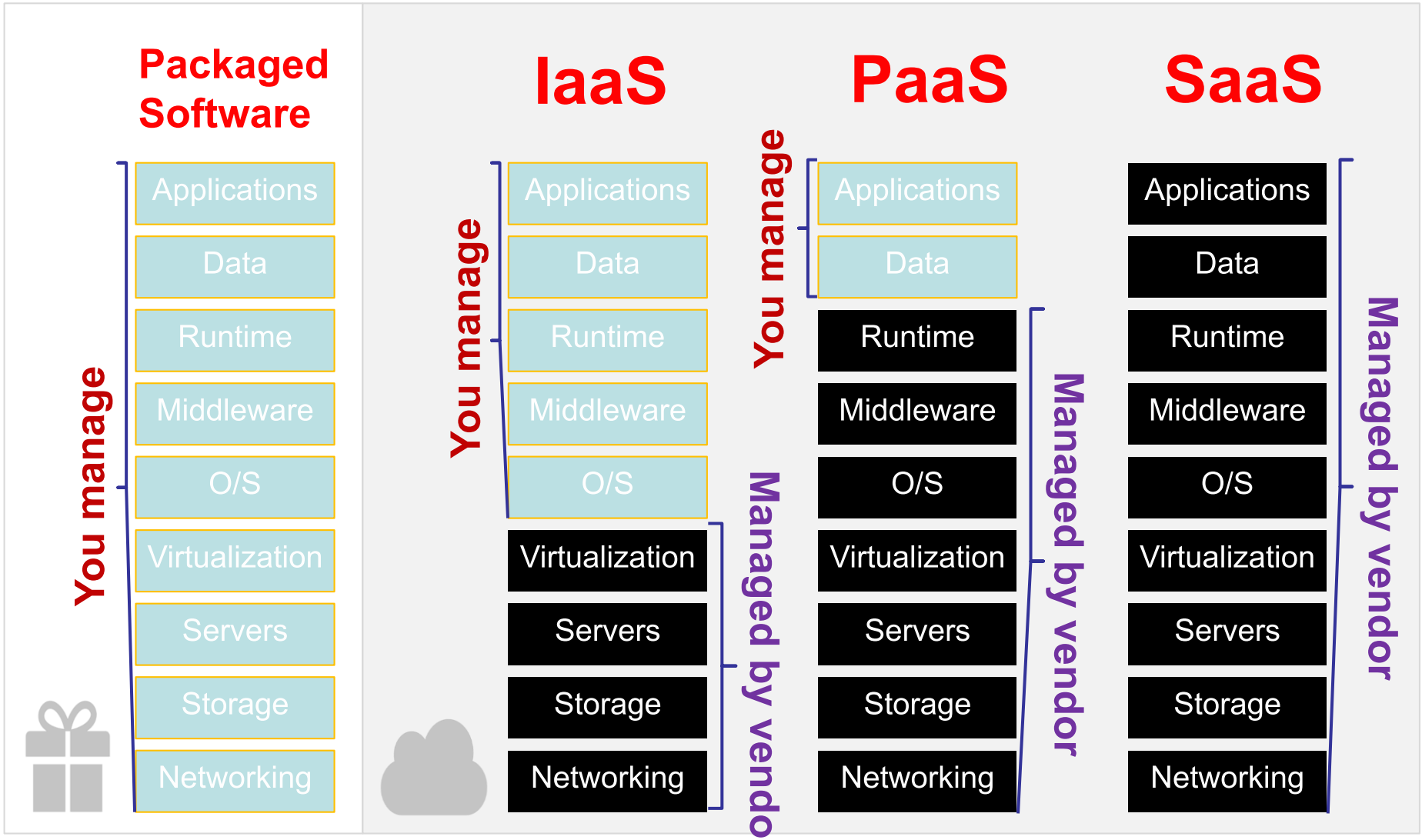
雲端運算的特性:

- 快速彈性：能因應要求調整資源規模大小。
- 計量服務：雲端服務各層次均由雲端供應者掌控與監管。
- 隨需自助服務：消費者可自行使用雲端服務，毋需與雲端供應者互動。
- 無所不在網路存取：雲端供應者服務可隨時在網路取用。
- 資源彙整：雲端供應者透過多重租賃模式服務消費者，依據消費者要求，來指派或重新指派實體及虛擬資源。

雲端運算的服務類型(Delivery Mode)

服務類型	IaaS 基礎設施即服務 (Infrastructure as a Service)	PaaS 平台即服務 (Platform as a Service)	SaaS 軟體即服務 (Software as a Service)
說明	專門提供設備、儲存空間、伺服器或專業，協助企業建置或使用雲端運算服務	為用戶提供一個託管平台，並提供開發環境，讓用戶可以將他們所開發的軟體部署在雲端	為用戶提供所需的應用服務安裝在網路上，客戶只要上網，就可使用，不需再下載至自己電腦而增加負擔
實際案例	Amazon EC2、Hicloud	Google Application Engine、Microsoft Azure	Google DOC、Salseforce.com CRM、Trend Micro SecureCloud

IaaS、PaaS、SaaS之區別



雲端運算的部署模型(Deployment Model)

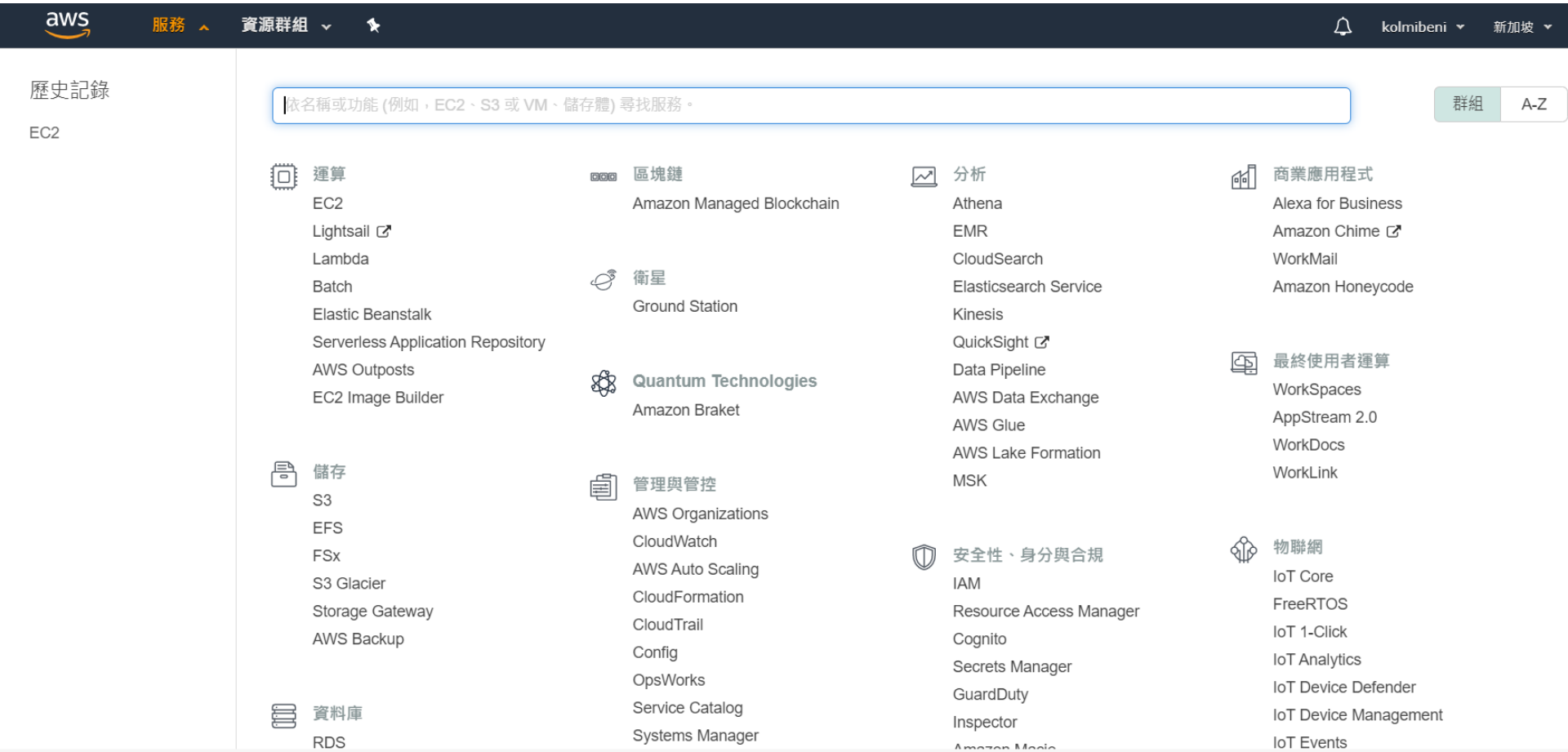
部署模型	公有雲 (Public Cloud)	私有雲 (Private Cloud)	混合雲 (Hybrid Cloud)
說明	使用者所需的服務皆由一個獨立的雲端供應商提供，同時也為這些使用者提供共享的雲端資源，適合中小型企業使用。	由某個企業獨立建構且使用的雲端運算環境，所有使用者皆是企業或組織內部的成員，適合大型企業如金融機構，製造業、政府機關等大型企業。	指私有雲和公有雲的混合，一般來說，大型企業等他們已經擁有一定規模的IT 基礎建設，他們可以將一些安全需求和可靠度較低的應用部署在公有雲上，減低自身IT基礎建設的負擔。

公有雲及私有雲比較

	公有雲	私有雲
優點	● 具有巨量且充沛的計算與儲存資源 ● 可以透過用多少付多少(Pay by Use)的計費模式，就能取得企業級的運算與儲存能力 ● 可以節省建置資訊基礎設施的昂貴成本 ● 可根據需求之大小，動態地調升或降低所需的運算與儲存資源，使得系統維運成本降至最低	● 面對的是內部的使用者、透過內部網路獲得和使用服務 ● 與公有雲相比，私有雲的規模較小，網路條件較好，使用者體驗較好，安全性較高
缺點	● 有些企業對於公有雲有安全性與資料保密性的疑慮	● 當出現突發性需求增長時，私有雲規模有限，難以像公有雲般快速地有效擴充，提供服務的成本也較高

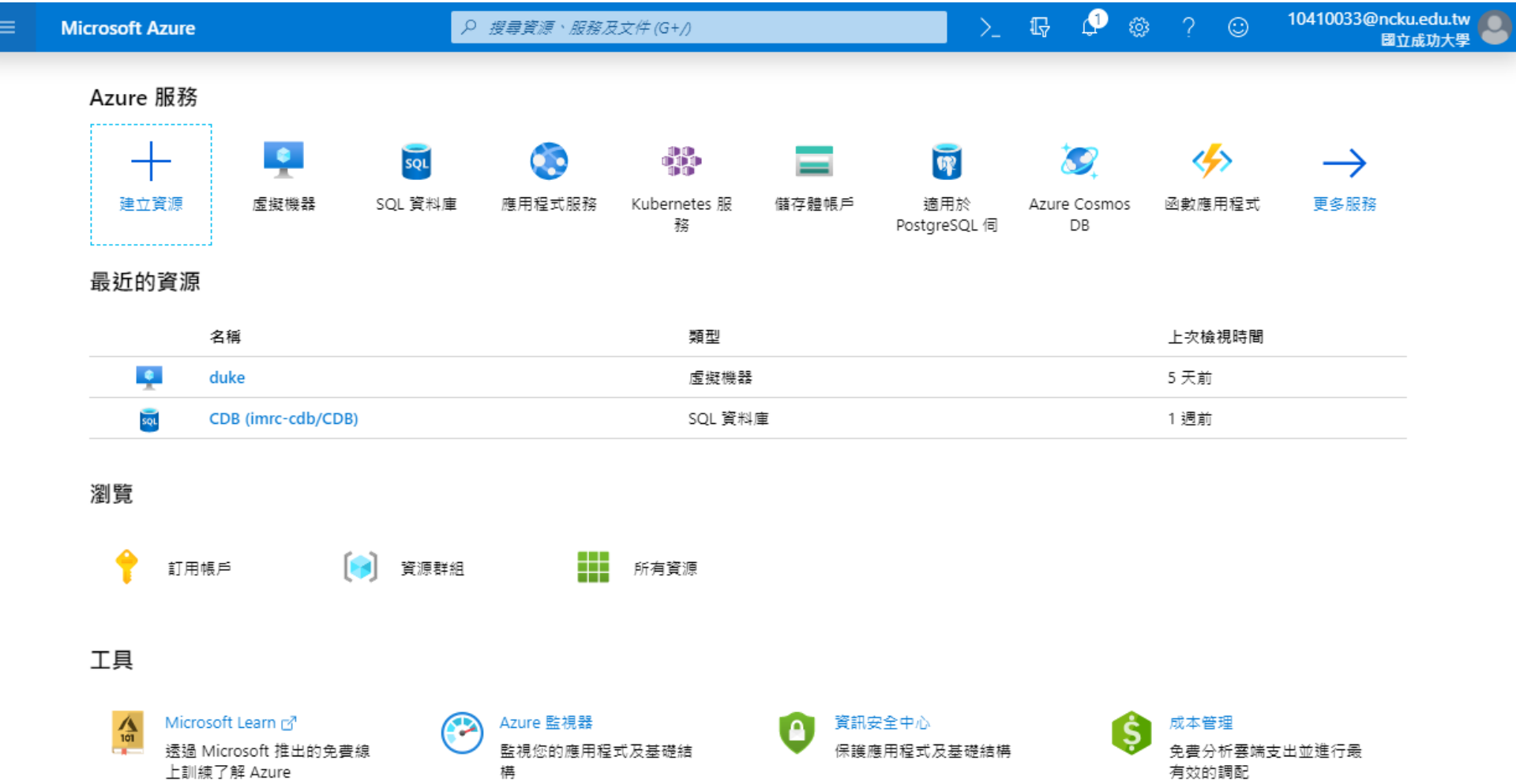
AWS公有雲帳號儀表板

- 雲端運算的特性：資源彙整、快速彈性、計量服務、隨需自助服務、無所不在網路存取



AWS公有雲帳號-EC2儀表板

Microsoft Azure公有雲帳號儀表板



Microsoft Azure全球資料中心

Global Footprint

Microsoft Azure協議的服務水平 (Service Level of Agreement)

- SLA =

  正常運行時間/一定期間(月/年)*100%

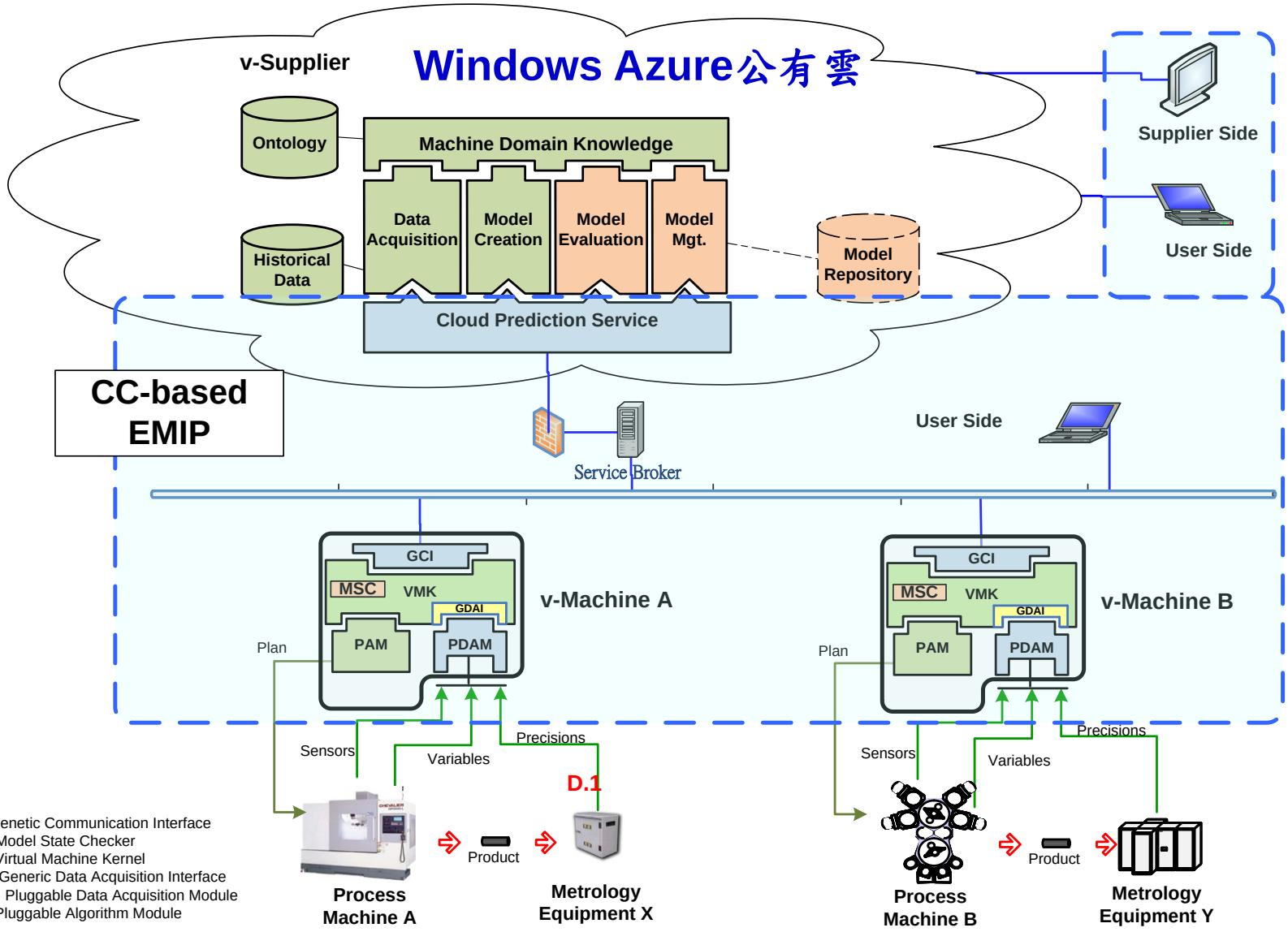
- 99.99% monthly SLA
  - Downtime Monthly: 4m 22s
- Pay only for what you use

每月上線時間百分比	服務折扣
< 99.99%	10%
< 99%	25%
< 95%	100%

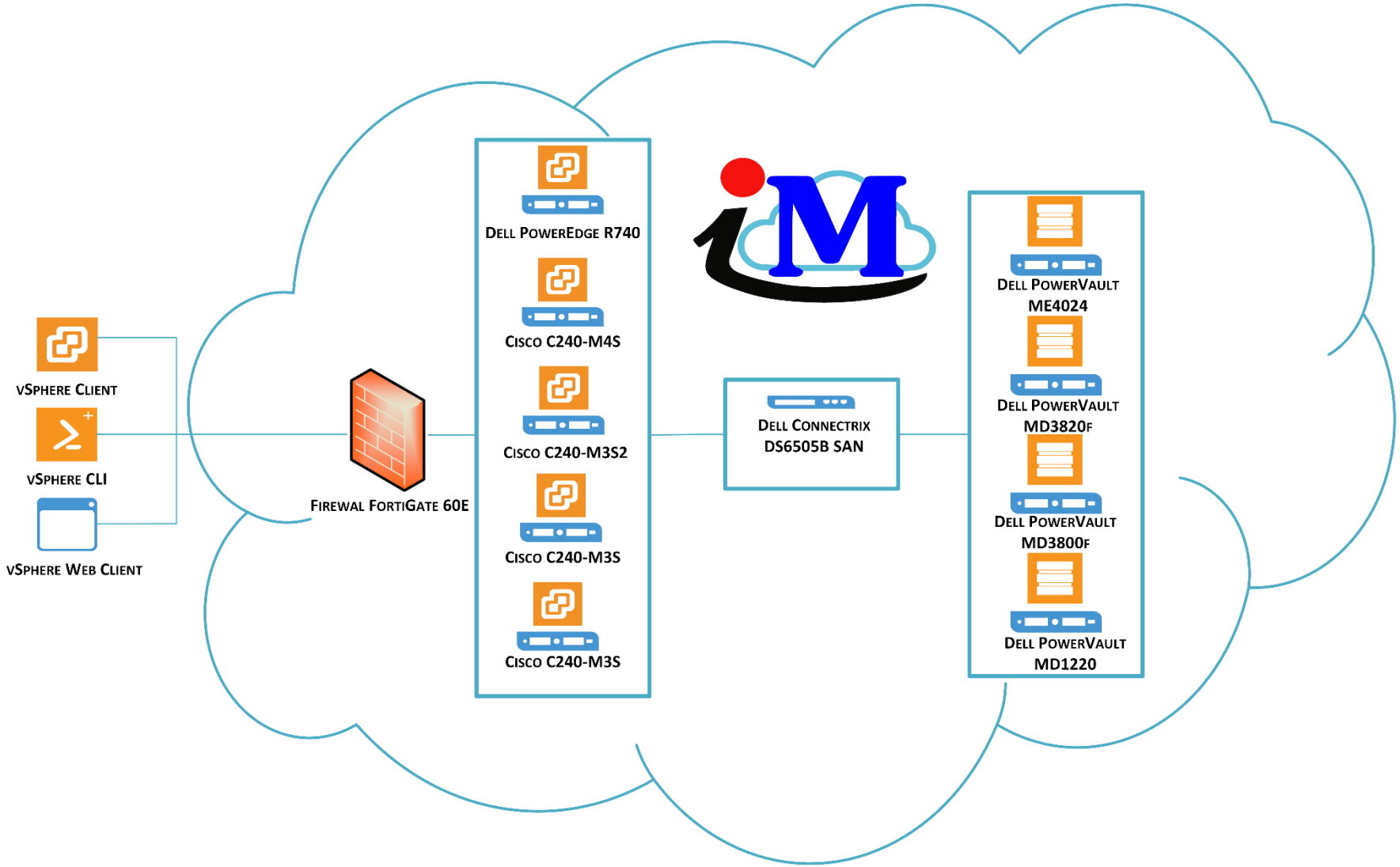
Azure公有雲可支援許多程式語言開發應用程式

Multiple languages

我們基於Azure發展之雲端機台監控系統

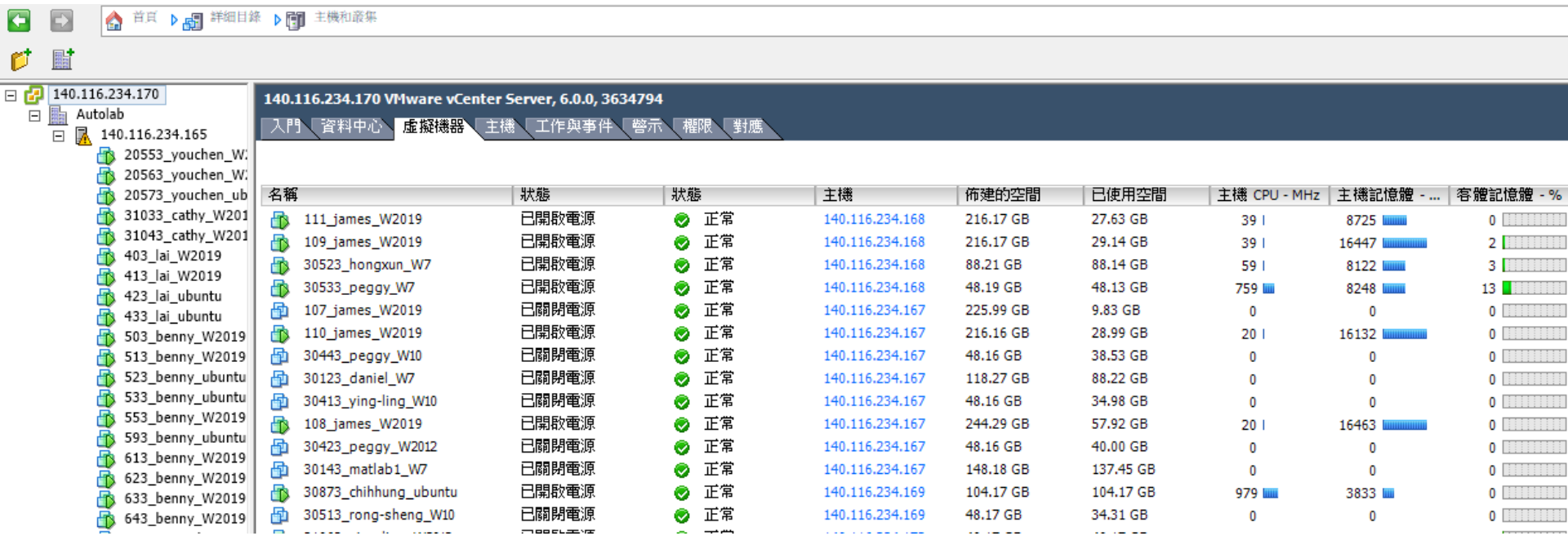


Private Cloud Setup



vSphere Client：Host and Cluster

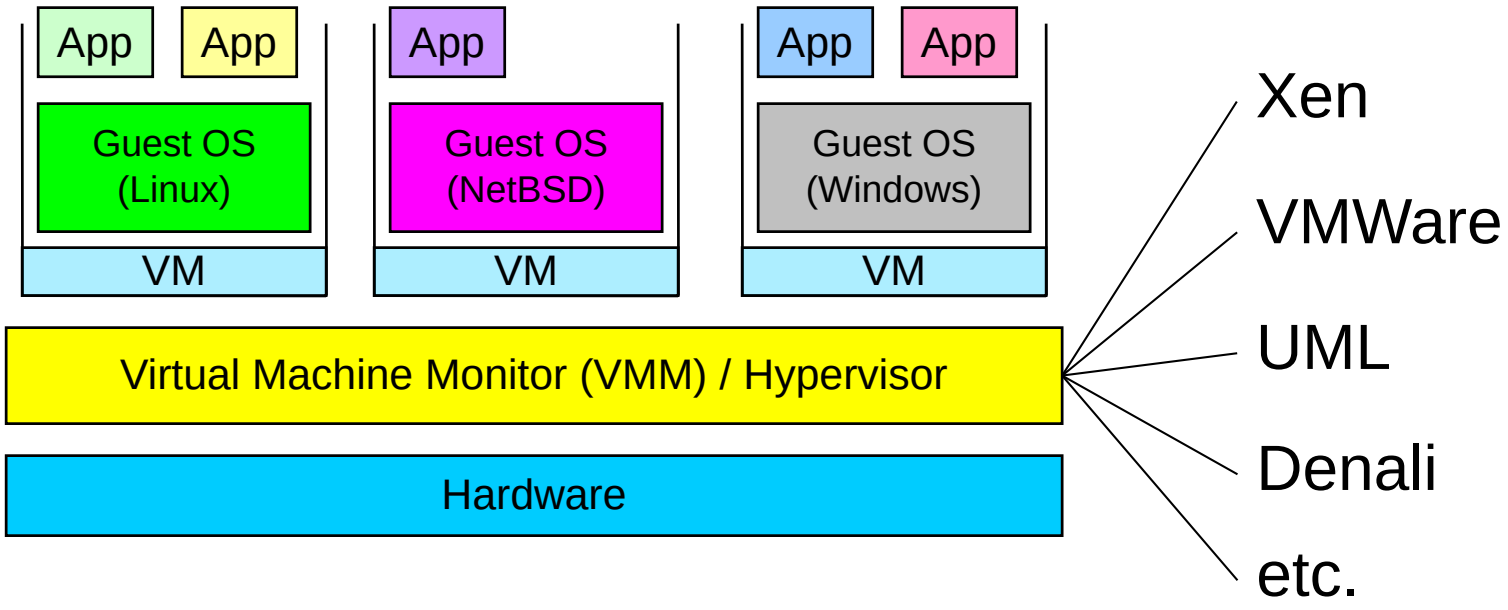
vSphere Client：Virtual Machine



名稱	狀態	狀態	主機	佈建的空間	已使用空間	主機 CPU - MHz	主機記憶體 - ...	客體記憶體 - %
111_james_W2019	已開啟電源	正常	140.116.234.168	216.17 GB	27.63 GB	39	8725	0
109_james_W2019	已開啟電源	正常	140.116.234.168	216.17 GB	29.14 GB	39	16447	2
30523_hongxun_W7	已開啟電源	正常	140.116.234.168	88.21 GB	88.14 GB	59	8122	3
30533_peggy_W7	已開啟電源	正常	140.116.234.168	48.19 GB	48.13 GB	759	8248	13
107_james_W2019	已關閉電源	正常	140.116.234.167	225.99 GB	9.83 GB	0	0	0
110_james_W2019	已開啟電源	正常	140.116.234.167	216.16 GB	28.99 GB	20	16132	0
30443_peggy_W10	已關閉電源	正常	140.116.234.167	48.16 GB	38.53 GB	0	0	0
30123_daniel_W7	已關閉電源	正常	140.116.234.167	118.27 GB	88.22 GB	0	0	0
30413_ying-ling_W10	已關閉電源	正常	140.116.234.167	48.16 GB	34.98 GB	0	0	0
108_james_W2019	已開啟電源	正常	140.116.234.167	244.29 GB	57.92 GB	20	16463	0
30423_peggy_W2012	已關閉電源	正常	140.116.234.167	48.16 GB	40.00 GB	0	0	0
30143_matlab1_W7	已關閉電源	正常	140.116.234.167	148.18 GB	137.45 GB	0	0	0
30873_chihhung_ubuntu	已開啟電源	正常	140.116.234.169	104.17 GB	104.17 GB	979	3833	0
30513_rong-sheng_W10	已關閉電源	正常	140.116.234.169	48.17 GB	34.31 GB	0	0	0

虛擬機器(Virtual Machines)

- **VM technology allows multiple virtual machines to run on a single physical machine.**



Performance: Para-virtualization (e.g. Xen) is very close to raw physical performance!

雲端運算(公有雲)的好處

- Lower computer costs
- Improved performance
- Reduced software costs
- Instant software updates
- Unlimited storage capacity
- Increased data reliability

雲端運算的缺點

- Requires a constant Internet connection
- Does not work well with low-speed connections
- Stored data might not be secure
- Stored data can be lost

Cloud-based AVM System

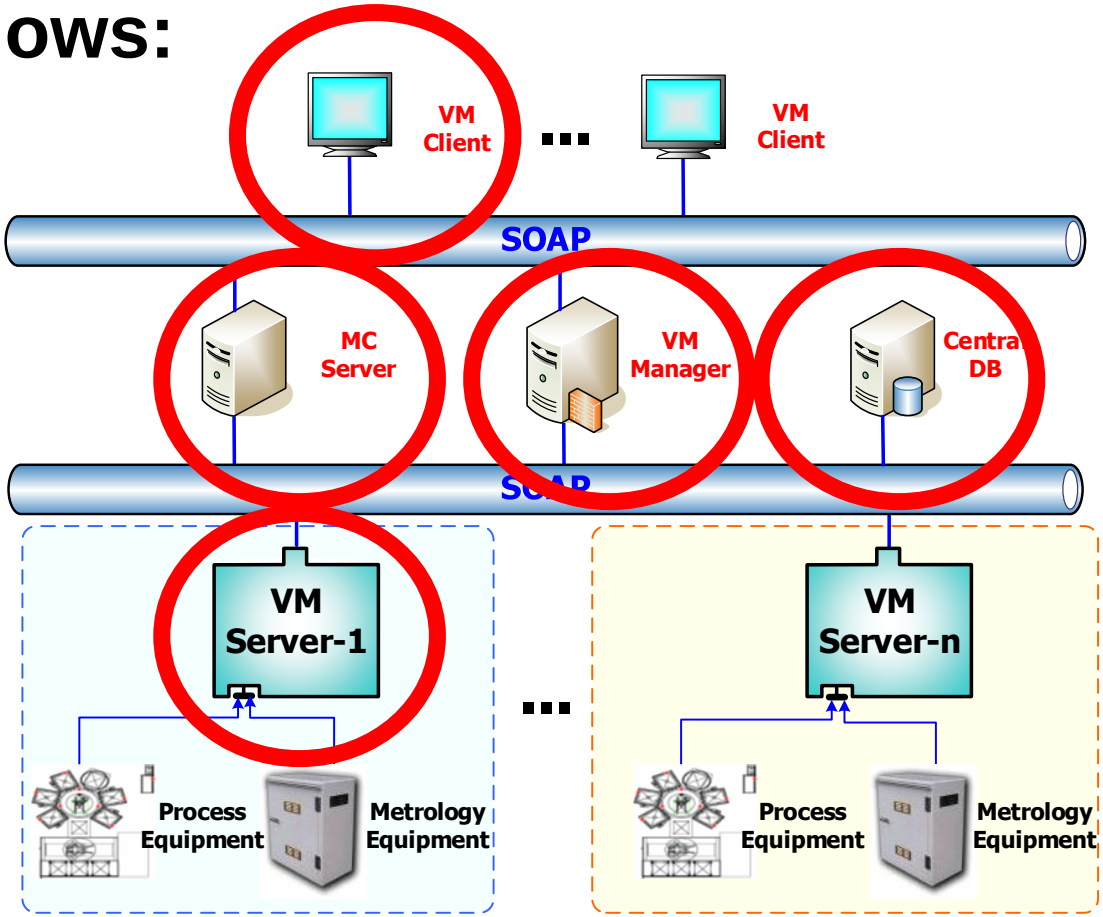
Background and Objectives

Introduction to Virtual Metrology (VM)

- **VM (Virtual Metrology) is able to predict the production quality of workpieces, such as wafers or glasses, using process data of the production equipment.**
- **VM can achieve real-time workpiece-to-workpiece quality monitoring.**
- **VM has become a viable approach to conduct online monitoring of production equipment and processes in high-tech manufacturing industries, such as Semiconductor, TFT-LCD, and Solar-Cell industries .**

Automatic Virtual Metrology (AVM) system

- The AVM system has been developed to facilitate factory-wide VM deployment as shown as follows:



Some Limitations of the Existing AVM System

Existing AVM systems have several limitations in a factory-wide deployment with a great number of equipment:

- **Incurring high hardware cost, occupying a great volume of shop-floor space, and needing complex management efforts.**
- **Deployment of VM servers needs to be conducted manually, which would be cumbersome and prone to error.**
- **The AVM system cannot automatically increase or decrease the number of VM servers on demand, usually resulting in over-provisioning (having idle VM servers) or under-provisioning (having insufficient VM servers).**

Limitations of the existing AVM System in Model Creation Functionality

- **No Support for Creating VM Models Simultaneously:**
  - Cannot allow multiple users to create their respective VM models simultaneously because of only using a MC server to undertake the MC tasks
- **No Support for Cross-Factory Usage:**
  - The existing AVM system was designed to be deployed in a factory so that it is confined to be used locally in a factory in terms of creating VM models
- **Lacking Robust Scheme for Processing MC Requests:**
  - Lack failover and load-balancing mechanisms for processing a large quantity of incoming MC requests.

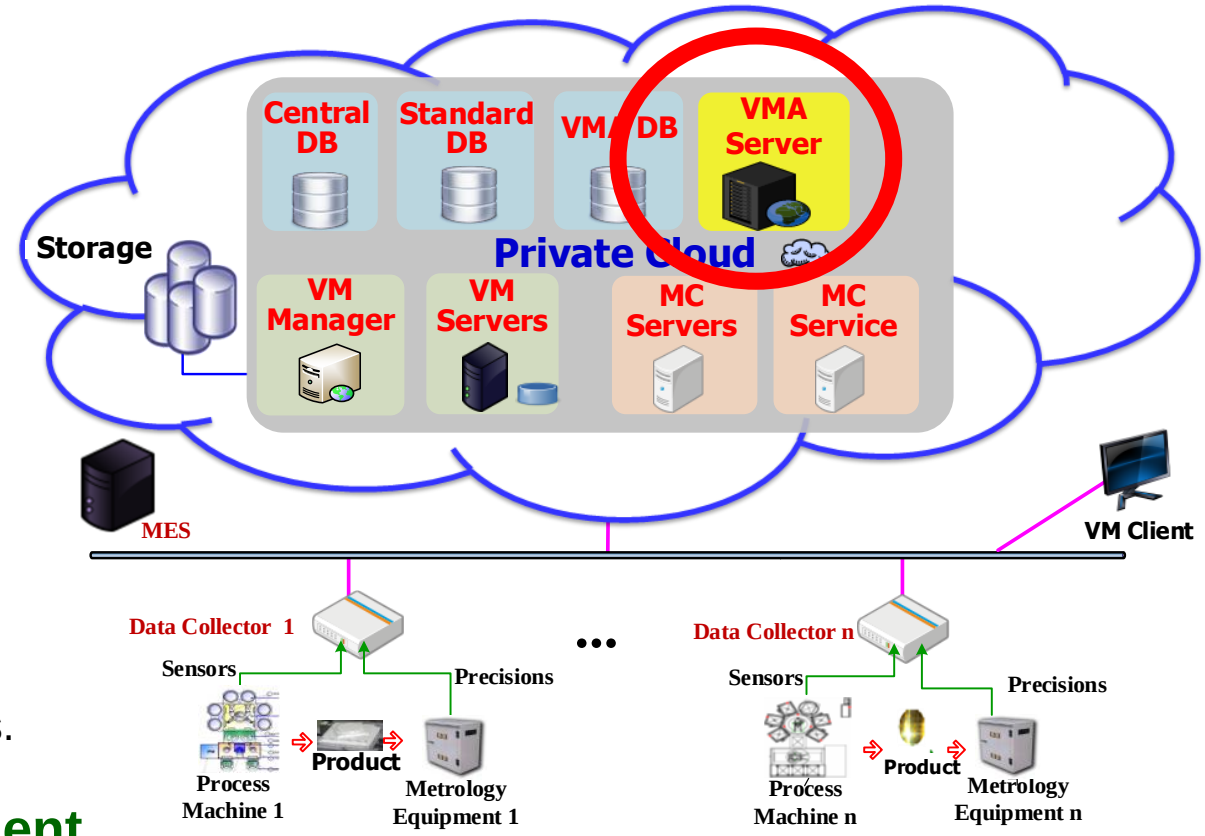
Objectives of this Work

- **Propose an approach of building cloud-based AVM systems, which could**
  - **effectively remedy the above-mentioned shortcomings of existing AVM systems in plant-wide deployment and in model creation functionality**
  - **keep the reworking efforts as little as possible to transform a AVM system into a cloud-based one.**

Architecture Design of Cloud-based AVM System

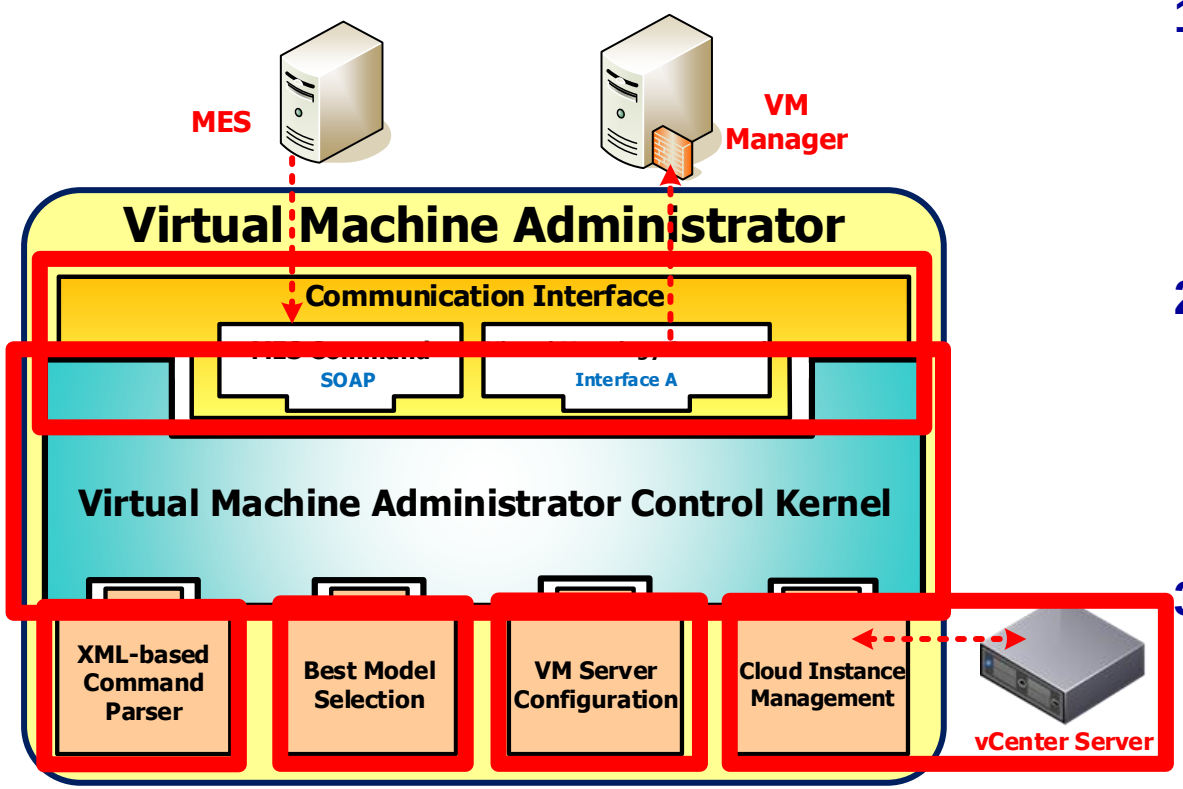
Architecture of the cloud-based AVM system

1. **Create a private cloud environment**
   VMware vSphere and its associated software, are used to create a private cloud environment.
2. **Virtualizing all servers**
   Leaving the original implementation codes of each server unchanged.
3. **Design an Virtual Machine Administrator (VMA)**
   Host and perform the major designed functional mechanisms.



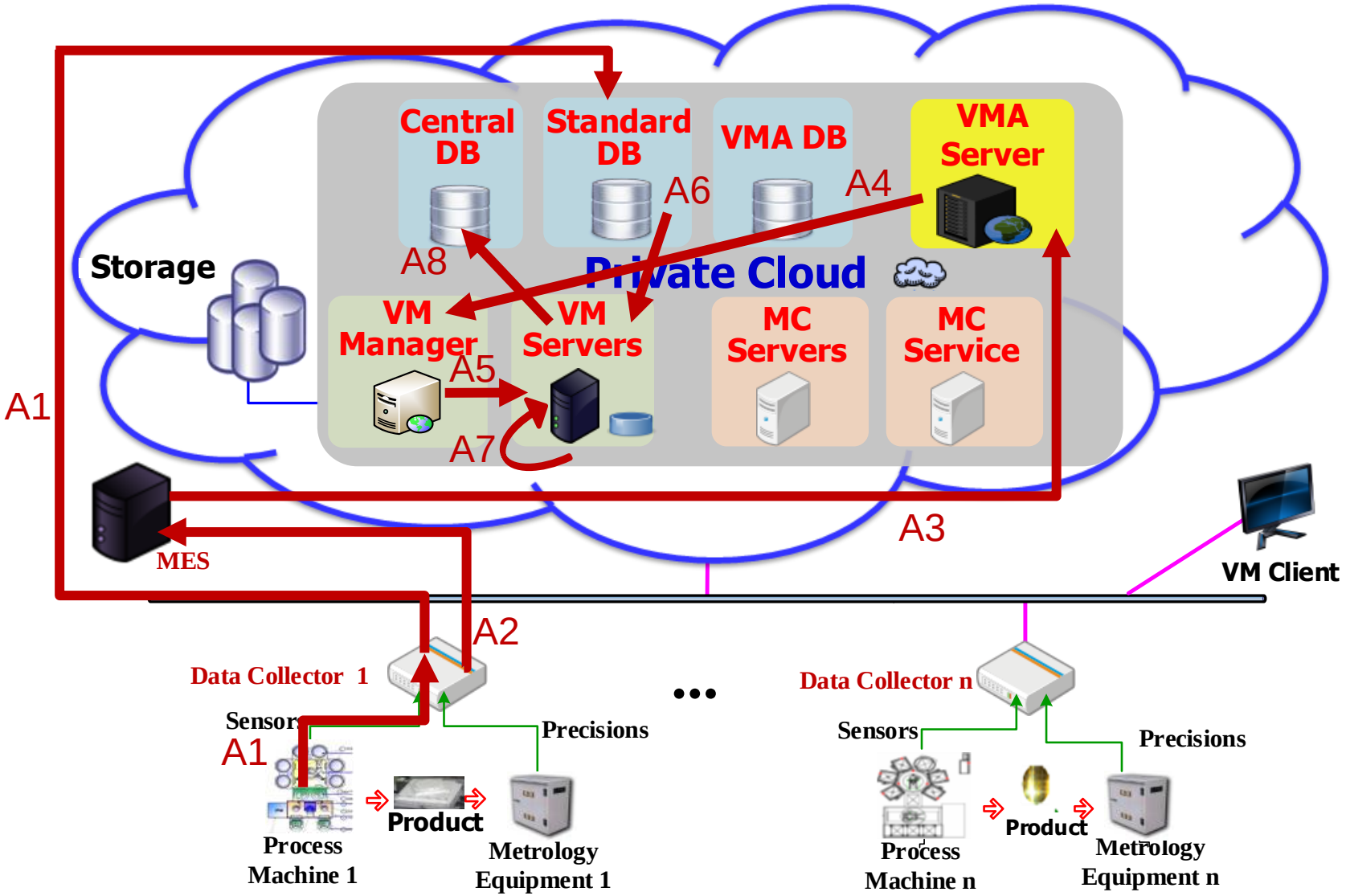
- **Automatic-Deployment**
- **Automatic-Scaling**
- **Automatic-Serving**

Framework Design of Virtual Machine Administrator



1. **VMA Control Kernel**
   Communicating with the MES and the VM manager, as well as handing messages and commands among functional components and performing functional mechanisms.
2. **Communication Interface**
   MES Command：
   Receive commands form MES.
   Virtual Metrology Command：
   Send commands to the VM manager to perform actions on the target VM server.
3. **Cloud Instance Management**
   Managing the virtual machines in the private cloud environment by using VMware vSphere API.
4. **VM Server Configuration**
   Automatically setting the configuration of the VM server.
5. **Best Model Selection**
   It is responsible for selecting the best-fit model from the Central DB.
6. **XML-based Command Parser**
   It is responsible for parsing the XML-based commands sent by the MES.

Operational flow of the cloud-based AVM system



Design of Core Functional Mechanisms

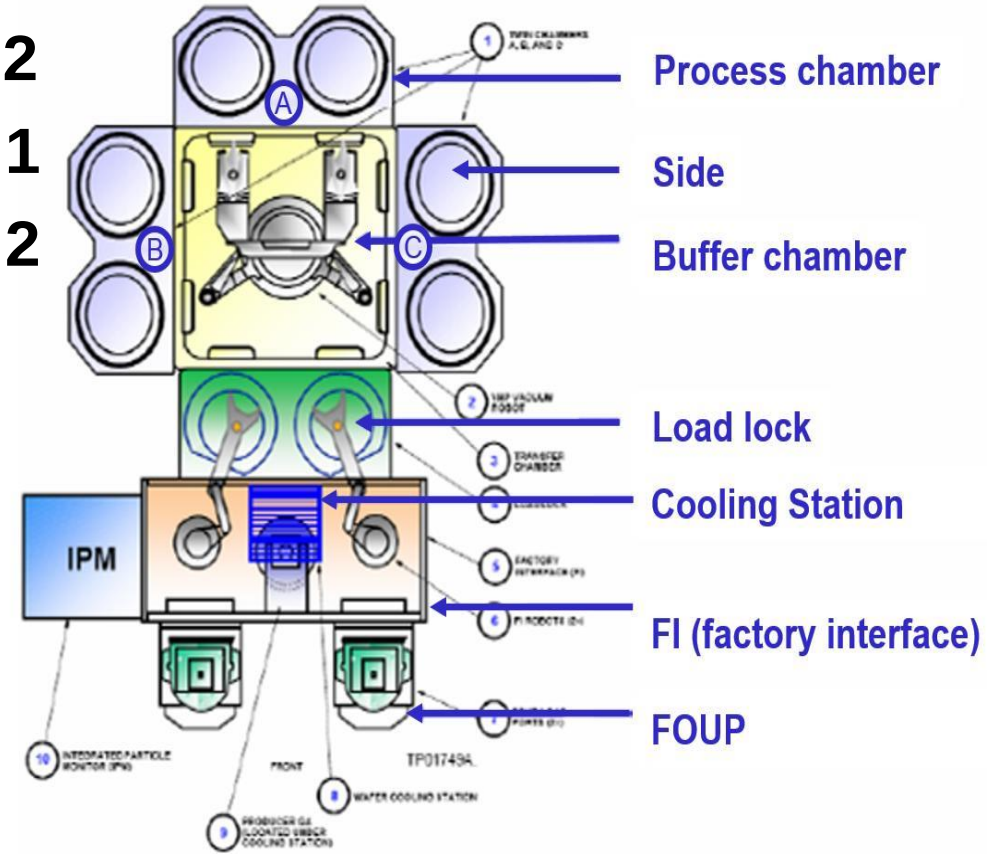
1. Design of Automatic-Scaling Mechanism
2. Design of Automatic-Deployment Mechanism
3. Design of Automatic-Serving Mechanism



Terminologies (1/2)

Combination Information of Equipment

- A01, Chamber A, Side 1
- A01, Chamber A, Side 2
- A01, Chamber B, Side 1
- A01, Chamber B, Side 2



Architecture of the ULKCVD Equipment A01

Terminologies (2/2)

- **Template Virtual Machine**

  A virtual machine installed with the VM server software, but lacking the combination information of equipment, prediction model, and IP.

- **Spare Virtual Machine**

  When a cloned virtual machine is turned on and assigned with an IP, it is called a spare virtual machine, which is ready to be configured become a virtual VM server.。

- **VM Server**

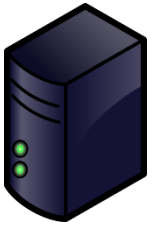
  When the spare virtual machine is equipped with prediction models and the combination information of equipment, it becomes a VM server.

Template Virtual Machine

[Clone]

Spare Virtual Machine

[prediction model+ combination information+IP]



VM Server

Schema Design of VMA Database

- **CombinationDEF Table:**
  Define the combination information.

Case	UMC	UMC	UMC	FATEK	FATEK
FieldName	Field1	Field2	Field3	Field1	Field2
Def	DEP_EQPID	DEP_CHAMBER	DEP_SIDE	STAGE	STEP

- **VMList Table: Store the statuses of virtual machines.**

ID	192.168.0.15	192.168.0.16	192.168.0.17	192.168.0.18
VMName	UMC-VMS-15	UMC-VMS-16	FATEK-VMS-17	ScaleOutVM
VMStatus	On	Off	On	On
LastestUsageTime	2013-08-12 17:21:47	2013-08-12 17:21:50	2014-02-12 14:15:32	
IPUsed	true	true	true	true
CombinationStatus	true	true	true	False
Field 1	A01	A01	OP1	
Field 2	A	A	2	
Field 3	1	2		
Case	UMC	UMC	FATEK	

active VM server VM server turned off spare virtual machine

Design of MES Command

- By parsing and extracting the information from the DEP_EQPID, DEP_CHAMBER, and DEP_SID elements, the VMA server is able to determine the combination for setting the VM server to serve the requested VM task.

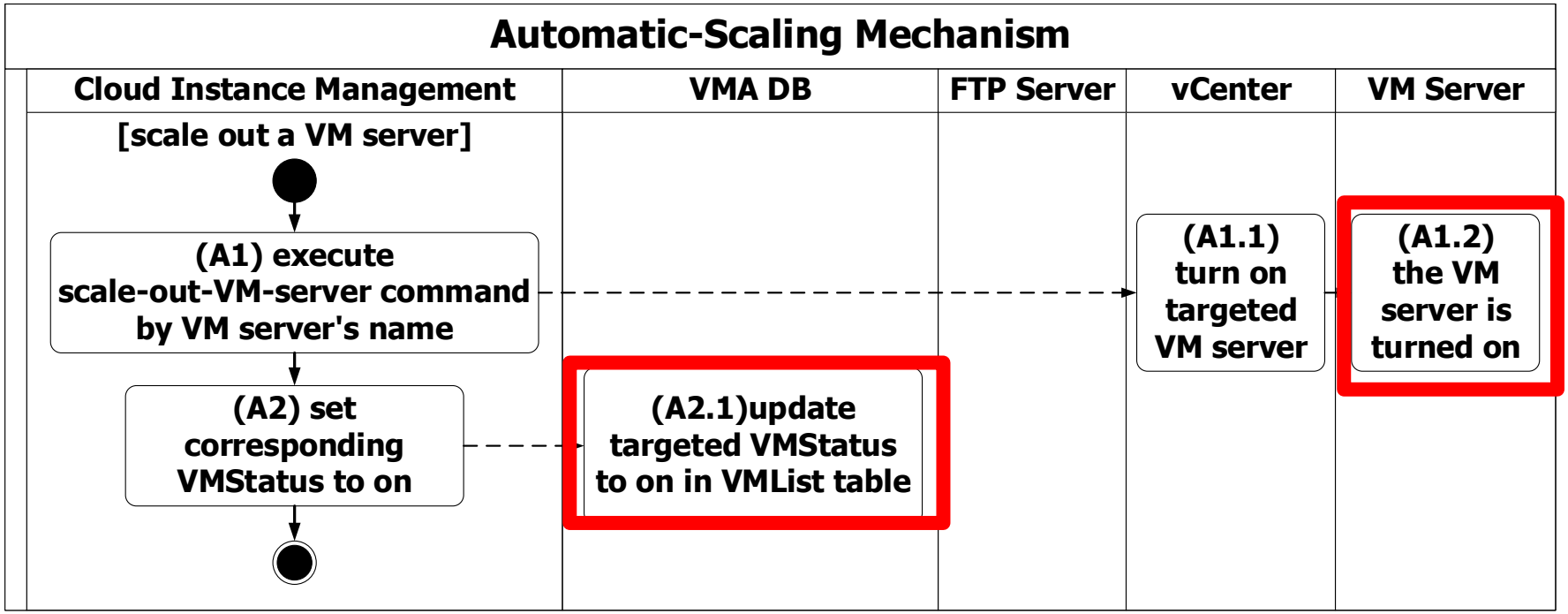
```
<?xml version="1.0" encoding="UTF-8"?>
- <message dir="VMC2VMM" wait="0" id="1" src="VMC  name="DownloadModel" xmlns:xsi:
  <command name="DownloadFileToVMS"/>
  - <parameters name="Combination">
    + <parameter name="PRODUCT">
    - <parameter name="DEP_EQPID">
        <value>/A09/</value>
      </parameter>
    - <parameter name="DEP_CHAMBER">
        <value>/A/</value>
      </parameter>
    - <parameter name="DEP_SIDE">
        <value>/1/</value>
      </parameter>
</message>
```

A

B

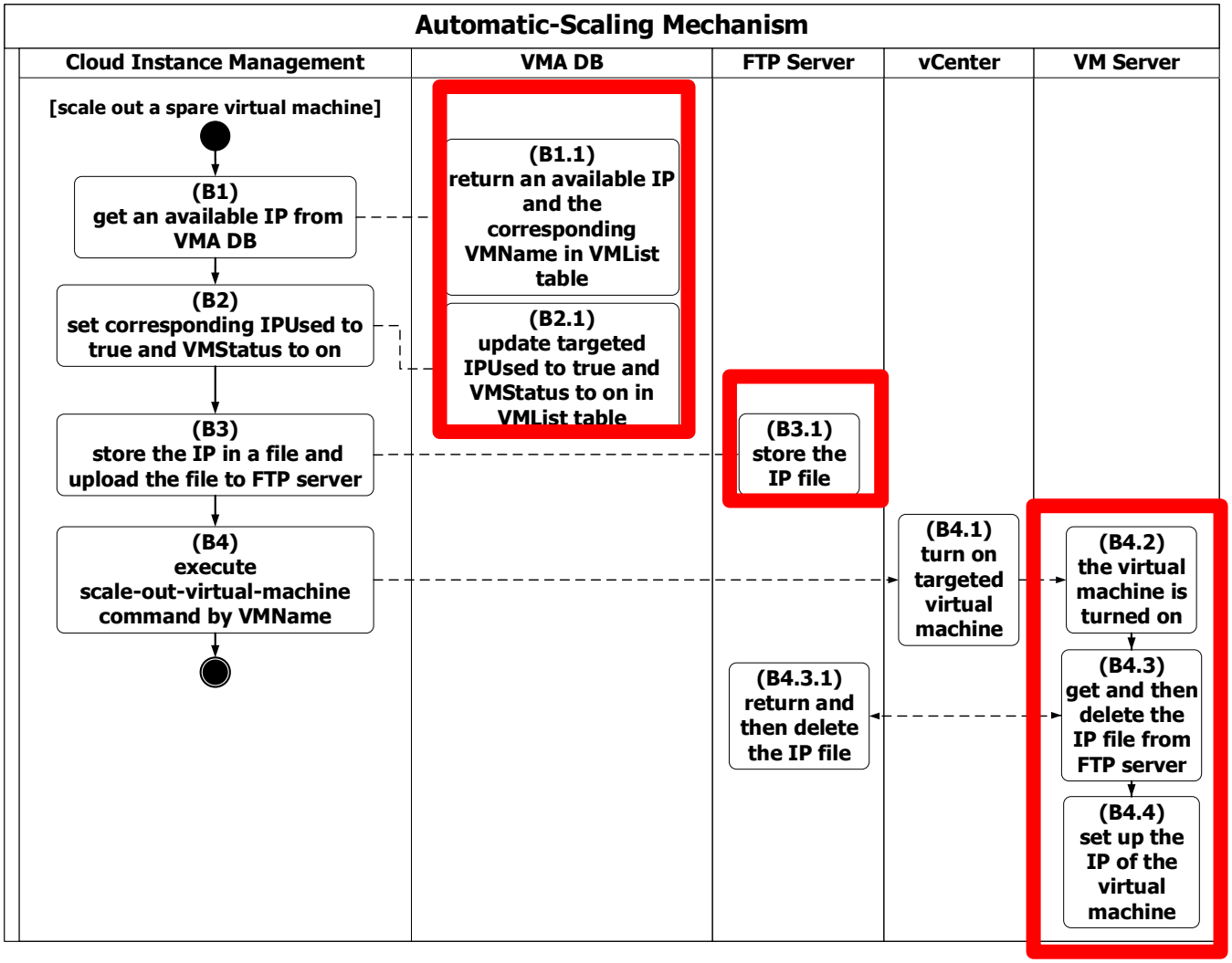
Design of Automatic-Scaling Mechanism (1/3)

- **Scale out a VM server:**
  to turn on a VM server



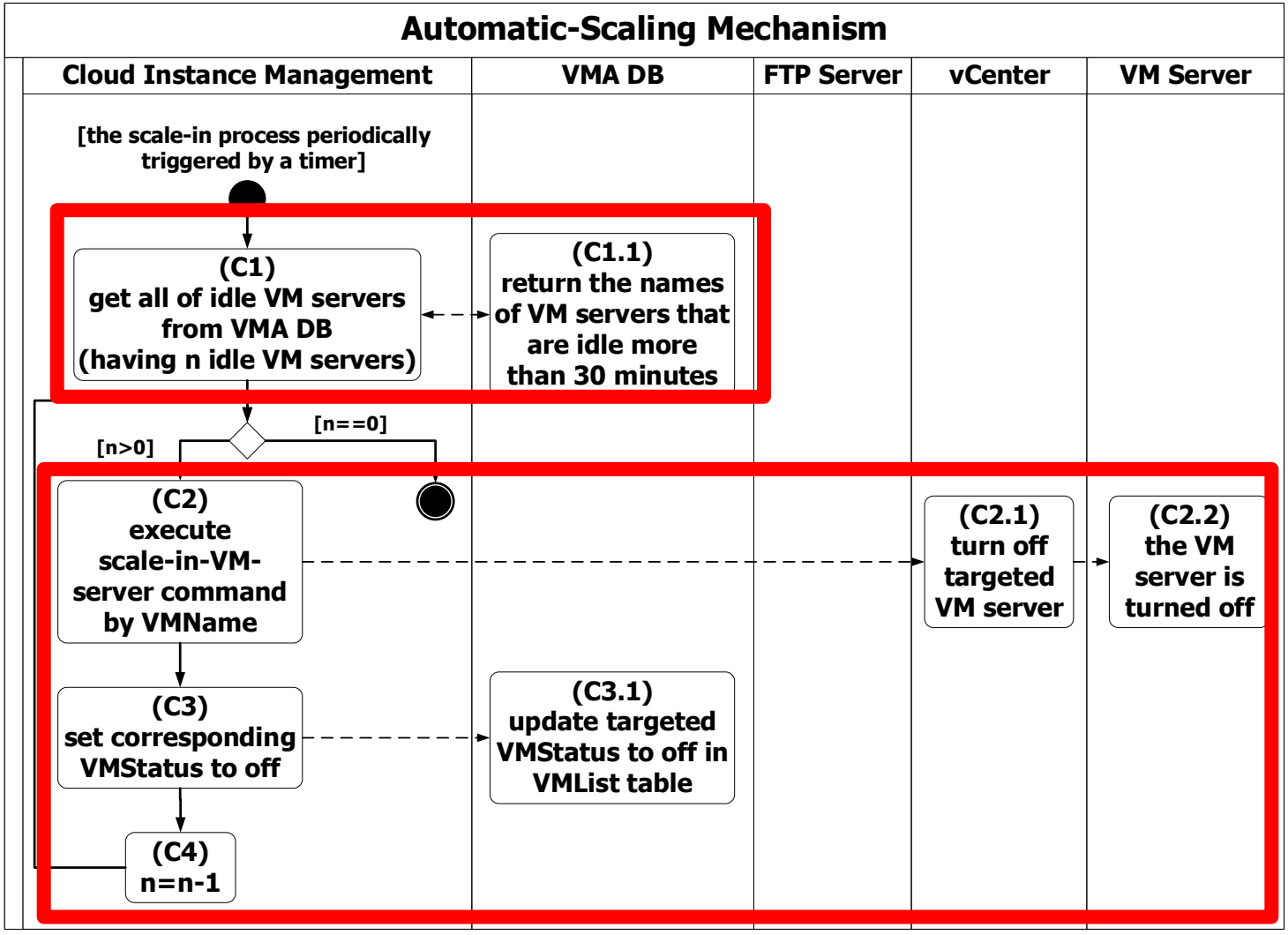
Design of Automatic-Scaling Mechanism(2/3)

- **Scale Out a Spare Virtual Machine:**
  to generate a new spare virtual machine



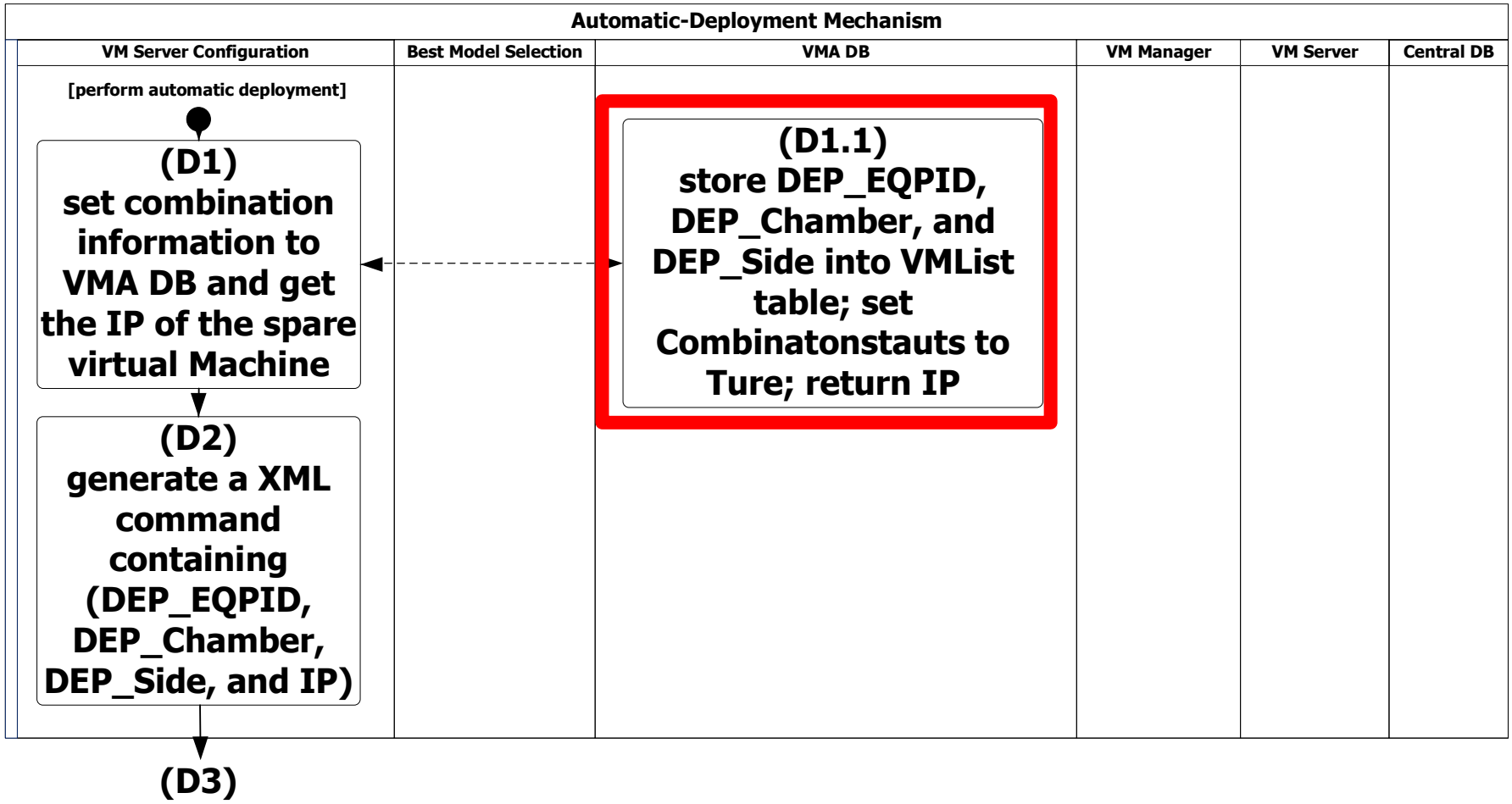
Design of Automatic-Scaling Mechanism(3/3)

- **Scale in a VM server:**
  to turn off a VM server (periodically triggered by a timer)

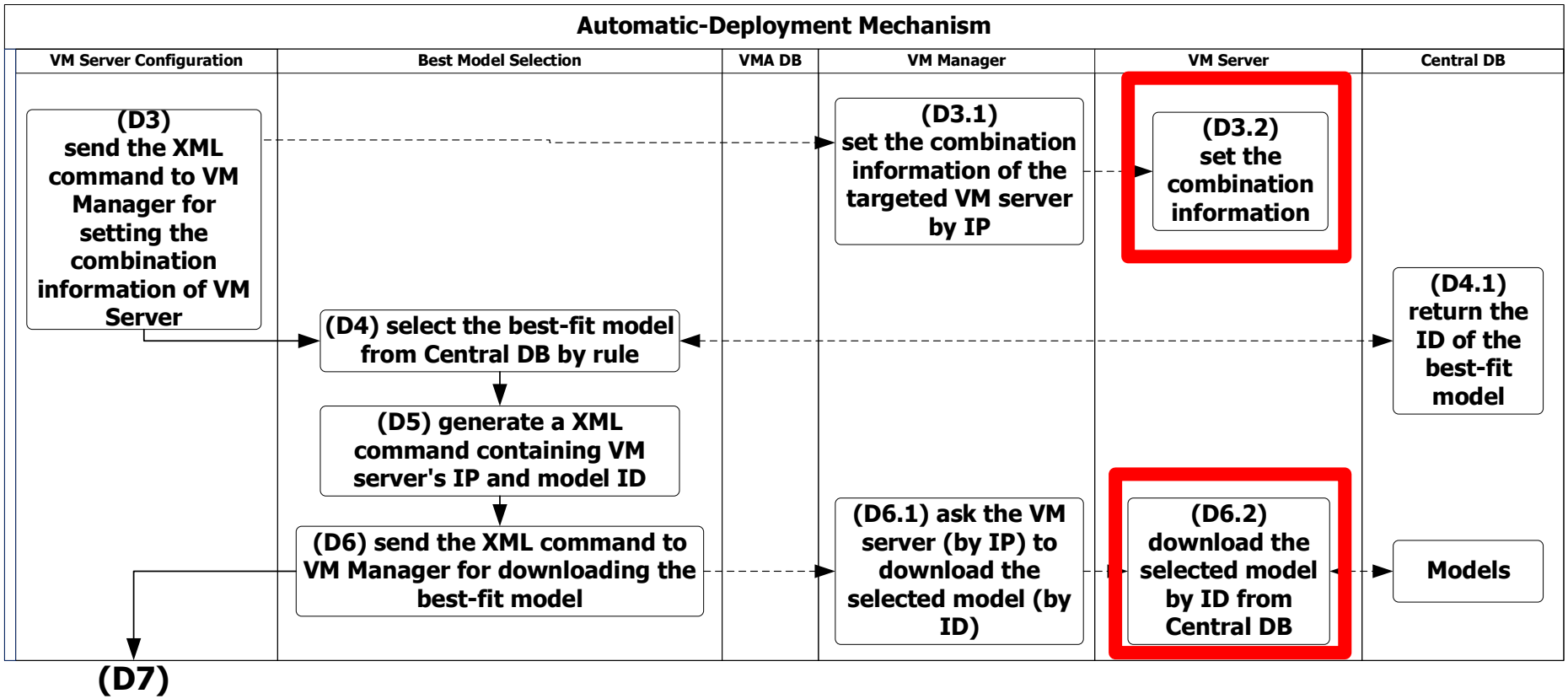


Design of Automatic-Deployment Mechanism(1/3)

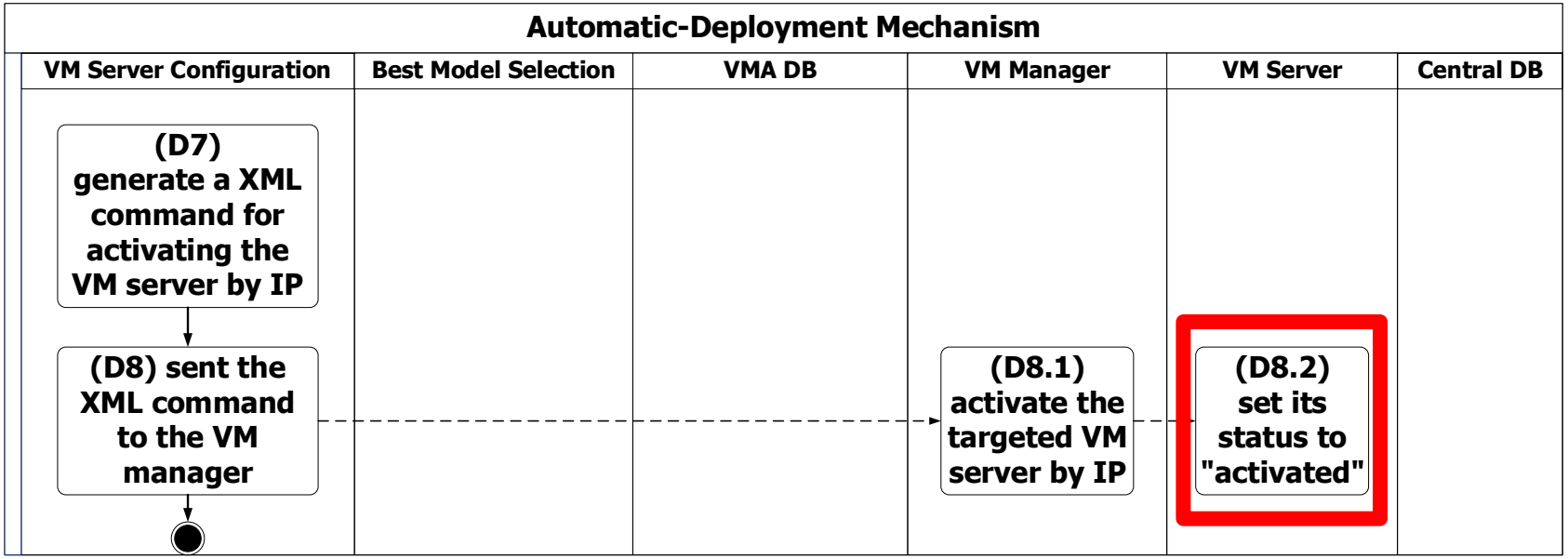
- This mechanism is to automatically transform a spare virtual machine into an active VM server:



Design of Automatic-Deployment Mechanism(2/3)

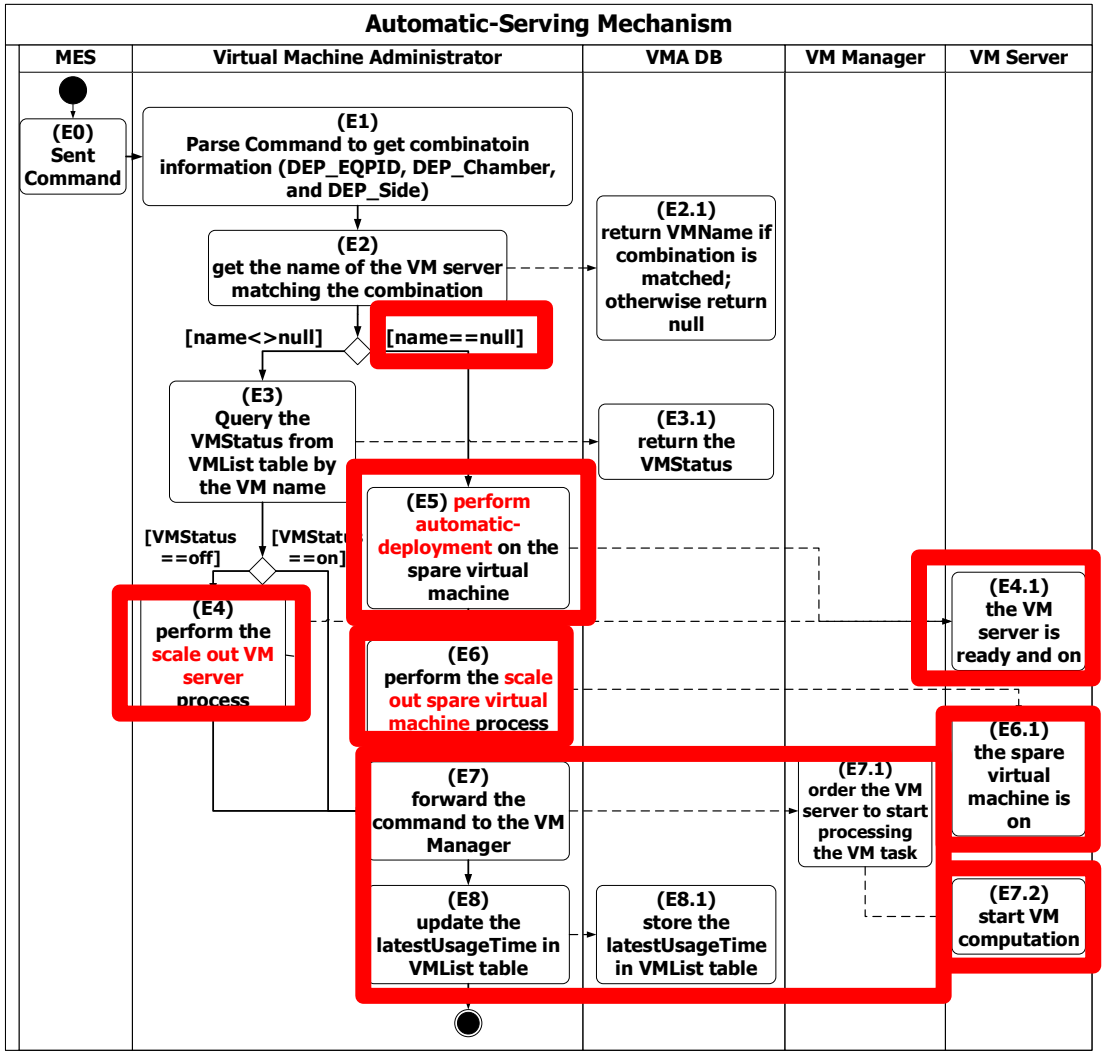


Design of Automatic-Deployment Mechanism(3/3)



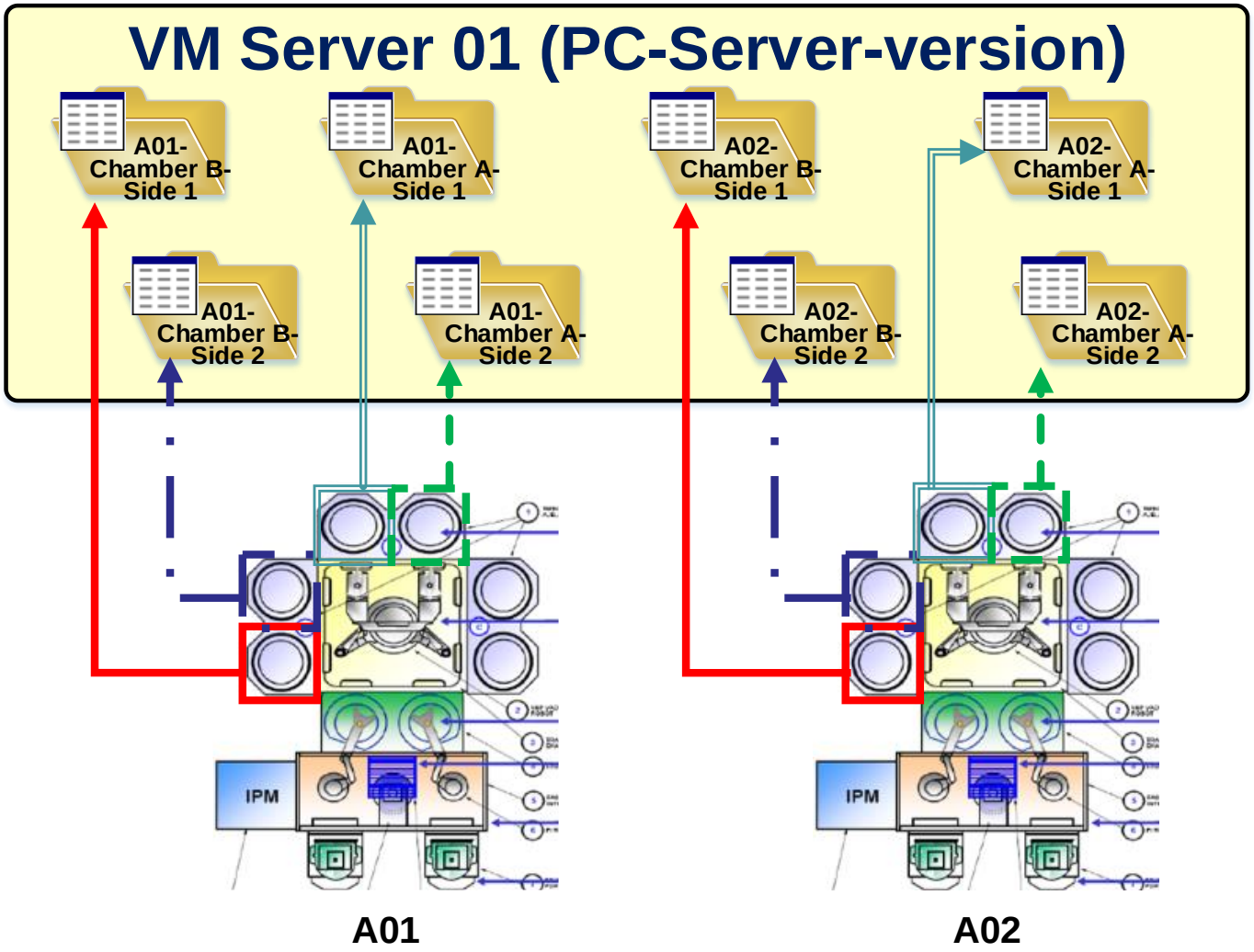
Design of Automatic-Serving Mechanism

- This mechanism is to automatically have a VM server to serve a requested VM task:



Case Study and Testing Results

Testing environment of PC-Server-version AVM system (1/2)



Testing environment of PC-Server-version AVM system (2/2)

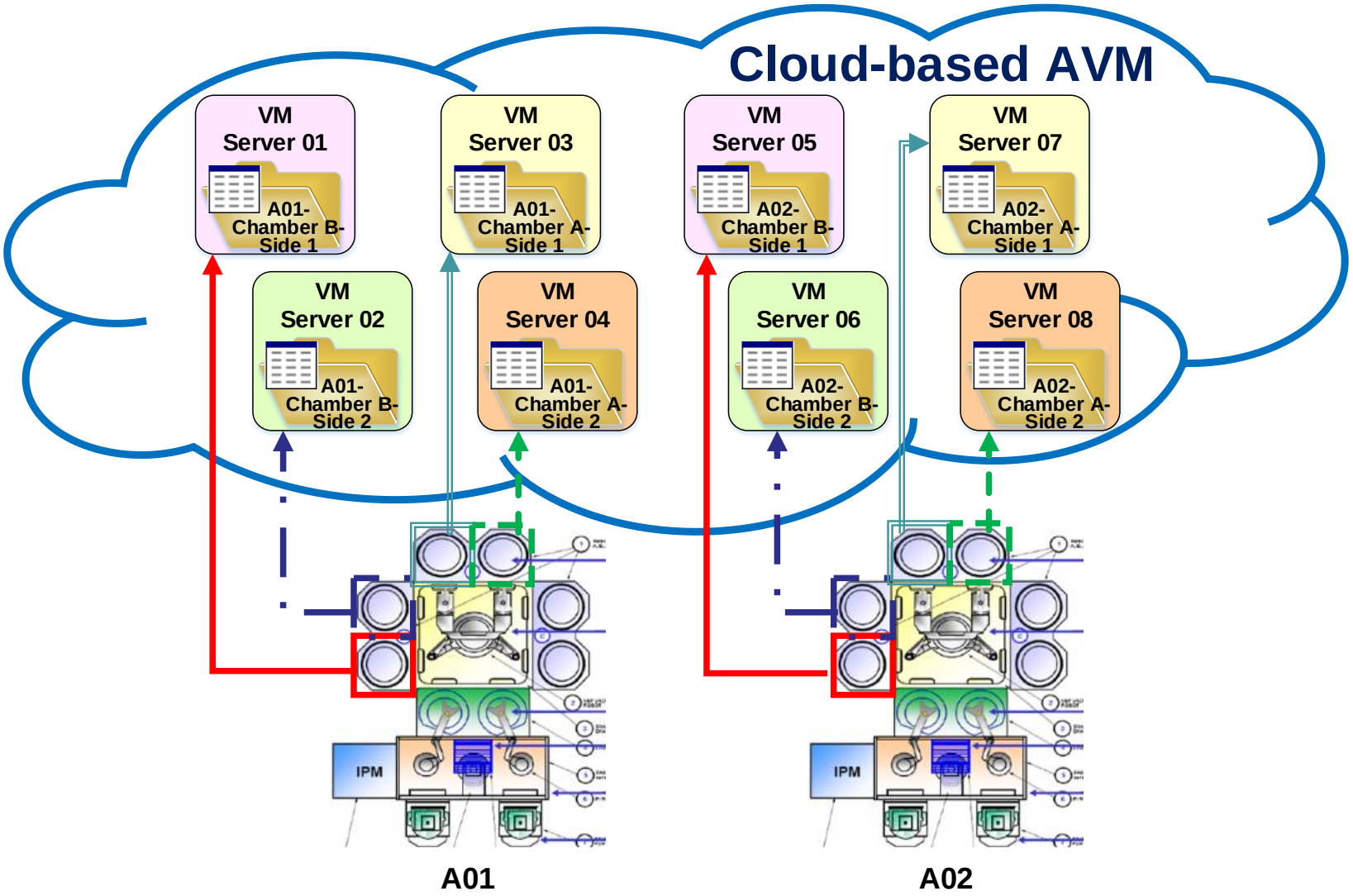
- **Physical Server Specifications:**

Computer	PC 1	PC 2
Software	VM Manager	VM Server
OS	Windows 7	Windows 7
CPU	Intel® Core™ i5-3450	Intel® Core™ i5-3450
Clock Speed	3.1 GHz	3.1 GHz
RAM	4 GB	4 GB

- **Test Data Information:**

DEP-EQPID	DEP-Chamber	DEP-Side	Time Period	Operation VM Server	Number of workpiece
A01	A	1	2012/10/15 09:30 ~ 2012/10/15 10:17	VM Server 01	10
A01	A	2	2012/10/15 09:00 ~ 2012/10/15 10:16		10
A01	B	1	2012/10/15 16:30 ~ 2012/10/15 16:40		10
A01	B	2	2012/10/15 09:00 ~ 2012/10/15 11:50		10
A02	A	1	2012/10/15 13:00 ~ 2012/10/15 16:20		10
A02	A	2	2012/10/15 09:00 ~ 2012/10/15 13:18		10
A02	B	1	2012/10/15 14:00 ~ 2012/10/15 17:20		10
A02	B	2	2012/10/15 09:00 ~ 2012/10/15 13:20		10
					Total 80

Testing environment of the cloud-based AVM system (1/2)



Testing environment of the cloud-based AVM system (2/2)

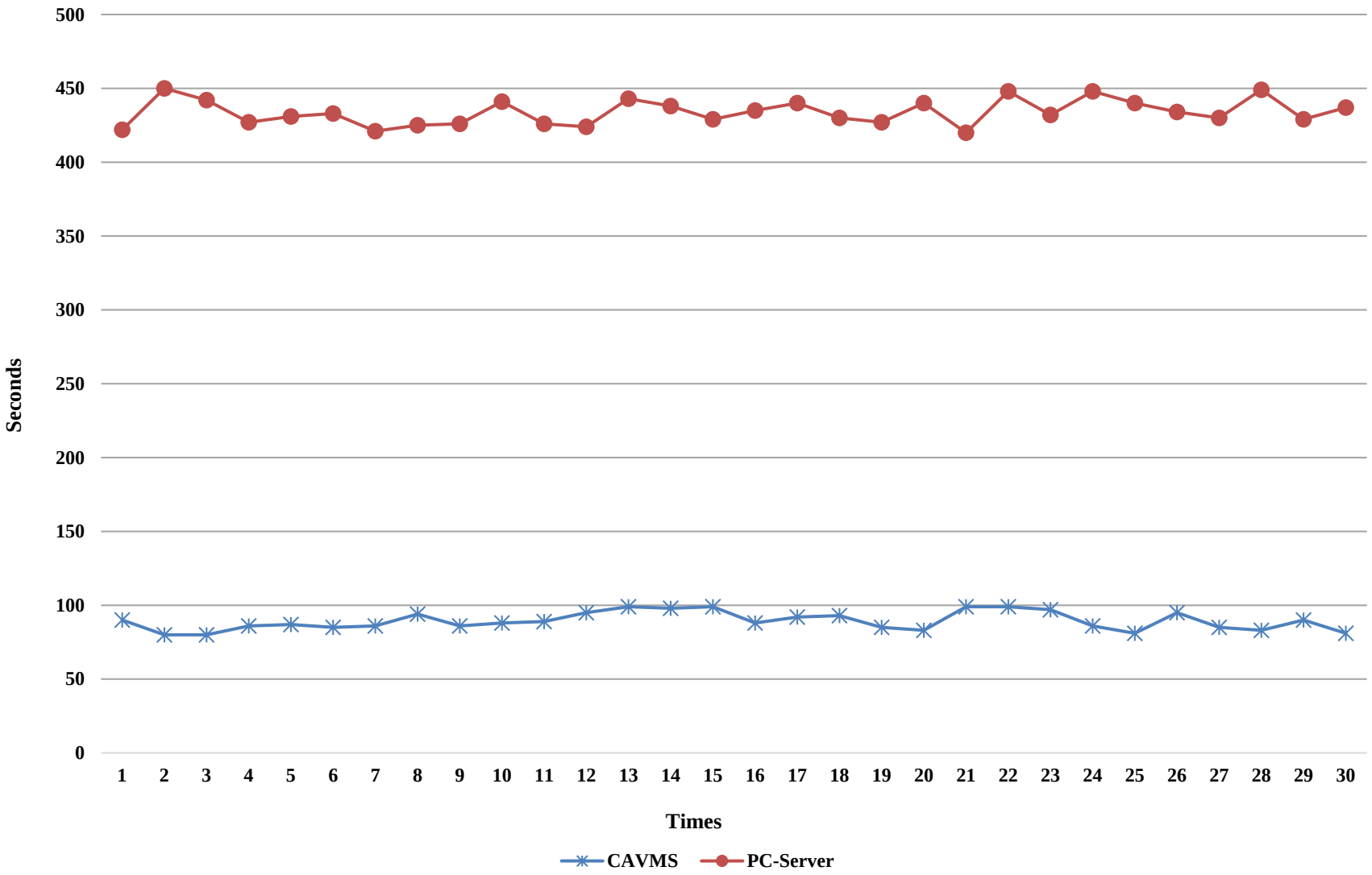
- **Virtual Machine Specifications**

Computer	Virtual Machine
OS	Windows 7
CPU	Intel® Xeon® E5-2625
Clock Speed	2.0 GHz
RAM	4 GB

- **Test Data Information**

Test Data Information					
DEP-EQPID	DEP-Chamber	DEP-Side	Time Period	Operation VM Server	Number of workpiece
A01	A	1	2012/10/15 09:30 ~ 2012/10/15 10:17	VM Server 01	10
A01	A	2	2012/10/15 09:00 ~ 2012/10/15 10:16	VM Server 02	10
A01	B	1	2012/10/15 16:30 ~ 2012/10/15 16:40	VM Server 03	10
A01	B	2	2012/10/15 09:00 ~ 2012/10/15 11:50	VM Server 04	10
A02	A	1	2012/10/15 13:00 ~ 2012/10/15 16:20	VM Server 05	10
A02	A	2	2012/10/15 09:00 ~ 2012/10/15 13:18	VM Server 06	10
A02	B	1	2012/10/15 14:00 ~ 2012/10/15 17:20	VM Server 07	10
A02	B	2	2012/10/15 09:00 ~ 2012/10/15 13:20	VM Server 08	10
					Total 80

Comparison of the execution time of predicting the production quality of 80 wafers



Comparison of Prediction Accuracy

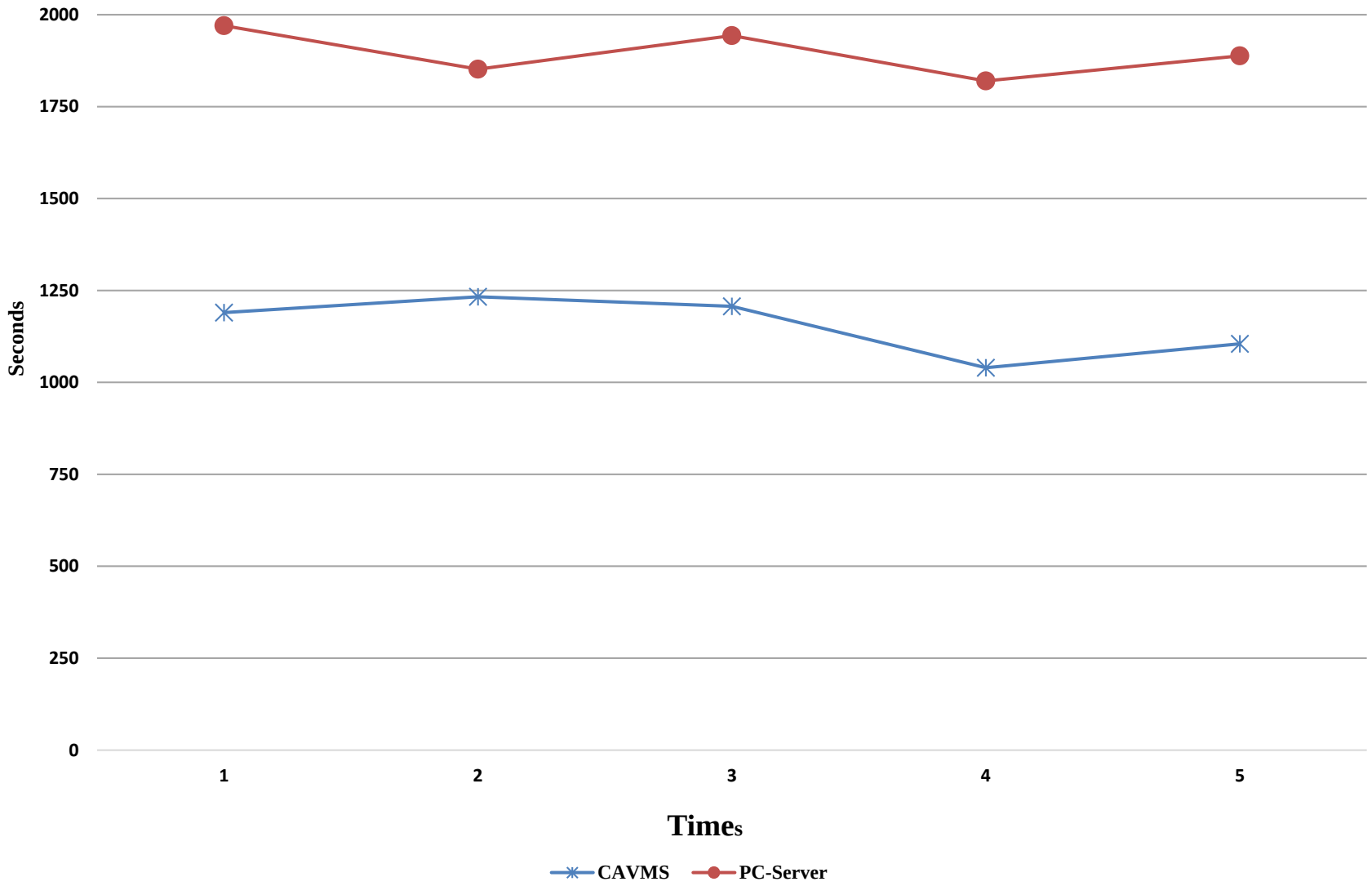
- **VM results of the PC-Server-based AVM system:**

NN	MAPE (%)	Standard Deviation
Phase I	0.3954	15.0091
Phase II	0.2894	14.6837

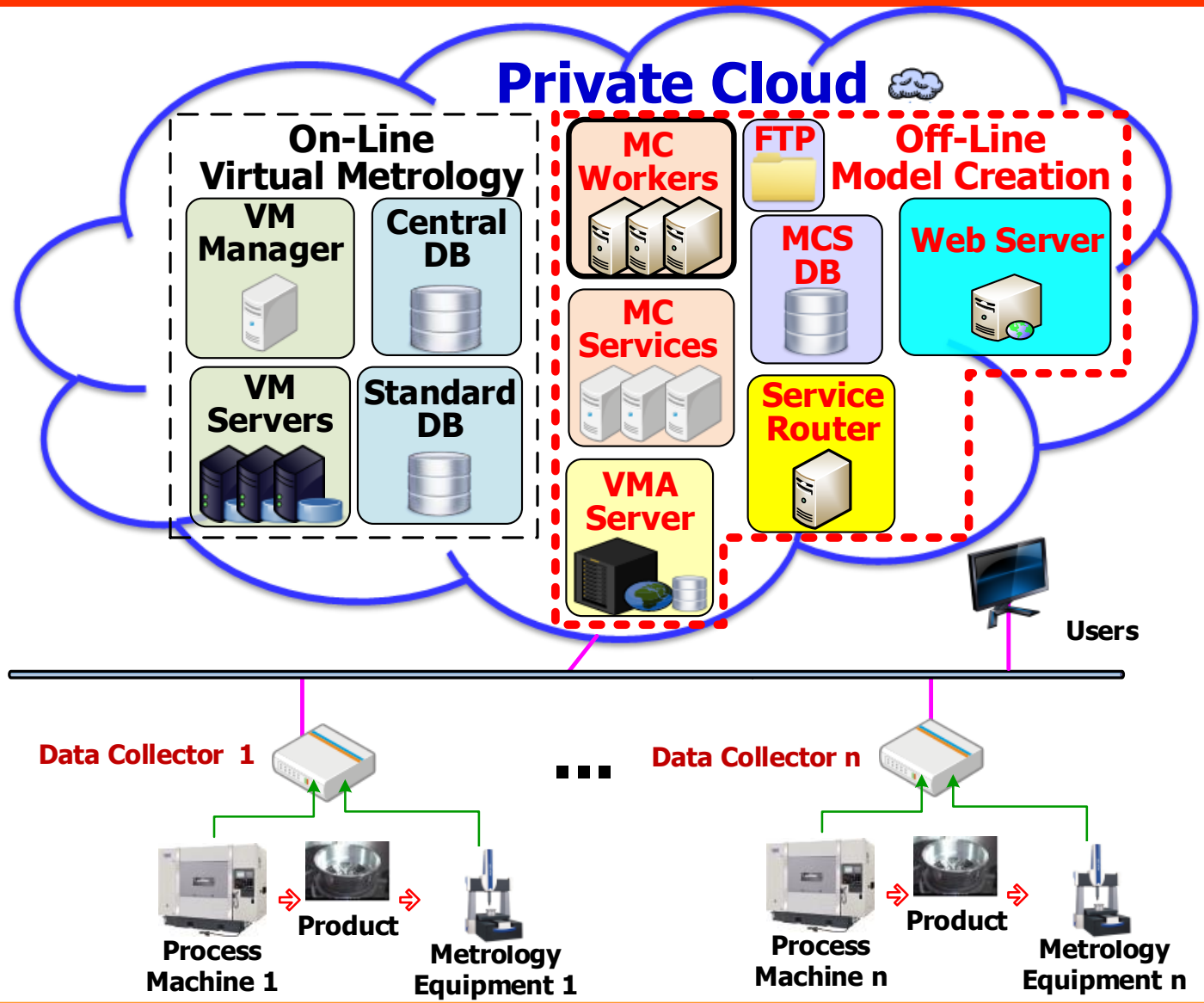
- **VM results of the cloud-based AVM system:**

NN	MAPE (%)	Standard Deviation
Phase I	0.3954	15.0091
Phase II	0.2894	14.6837

Comparison of the time to deploy 8 VM servers (automatic deployment mechanism vs. manual deployment)



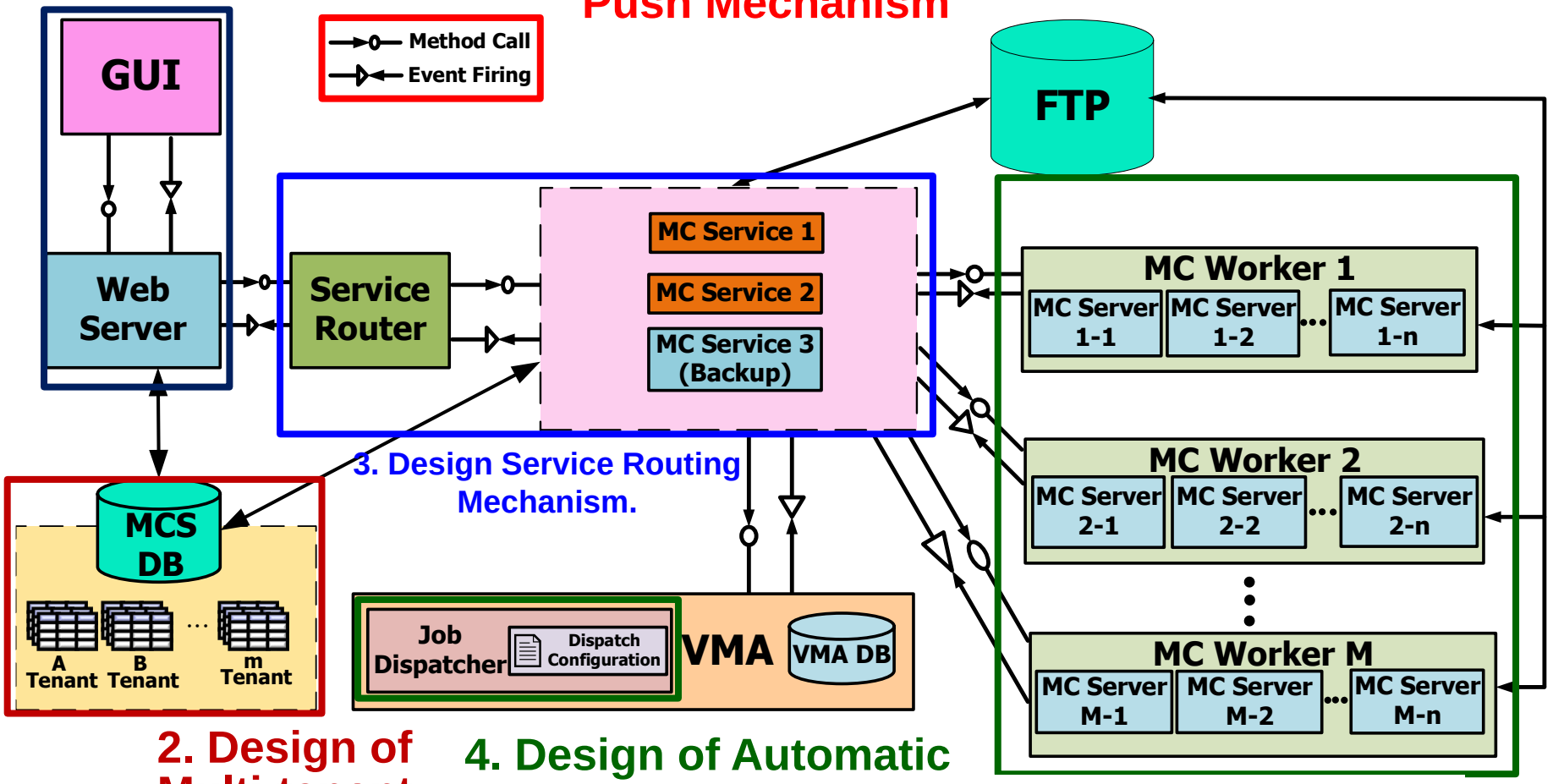
Cloud AVM system architecture based on private cloud technology



Multi-tenant Model Creation Services for AVM based on Private Cloud

1.Rewrite GUIs by HTML5 technology.

5. Design of Active Push Mechanism



3. Design Service Routing Mechanism.

2. Design of Multi-tenant Mechanism

4. Design of Automatic Job Dispatching Mechanism

4. Design of Multi Processing Mechanism

Testing Data and Virtual Machine Specifications

- Equipment and workpiece

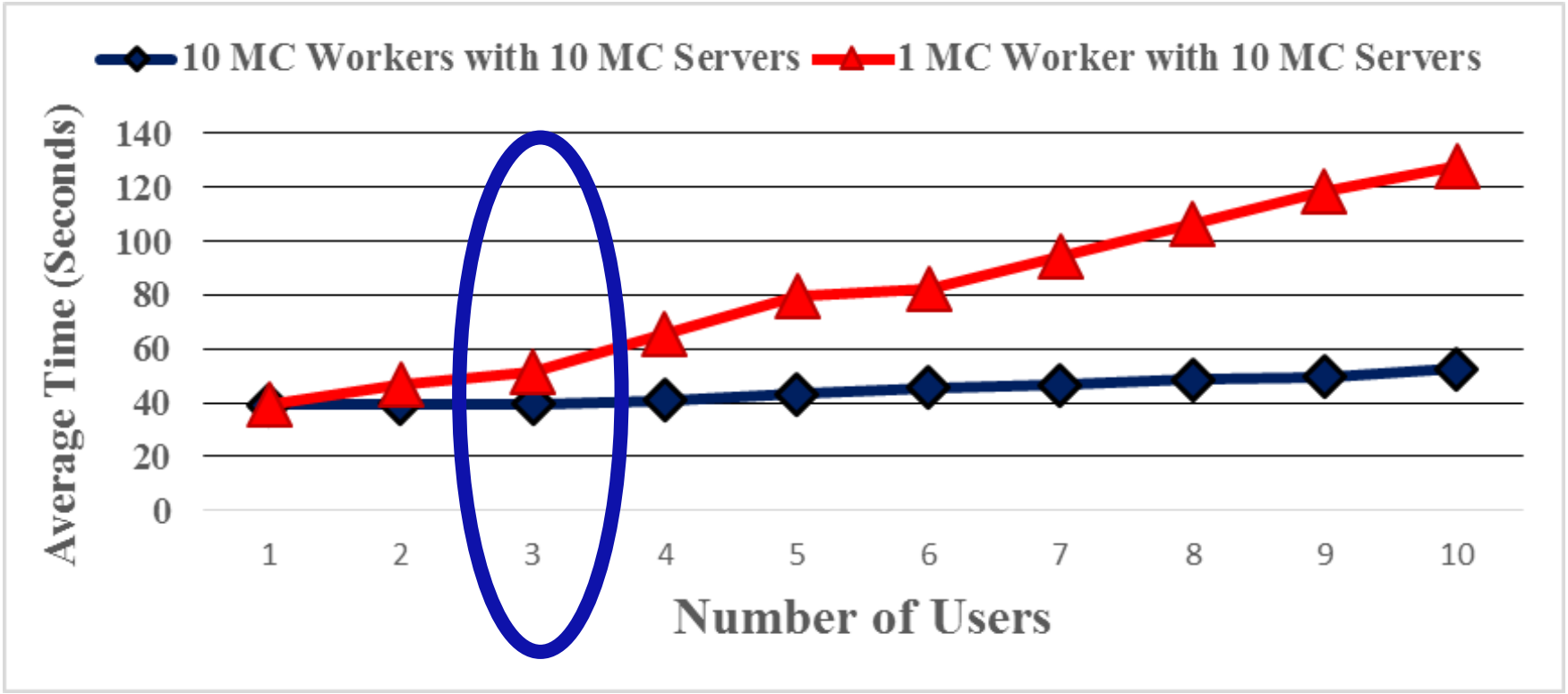
WVL-F24 Lathe

- V Specifications

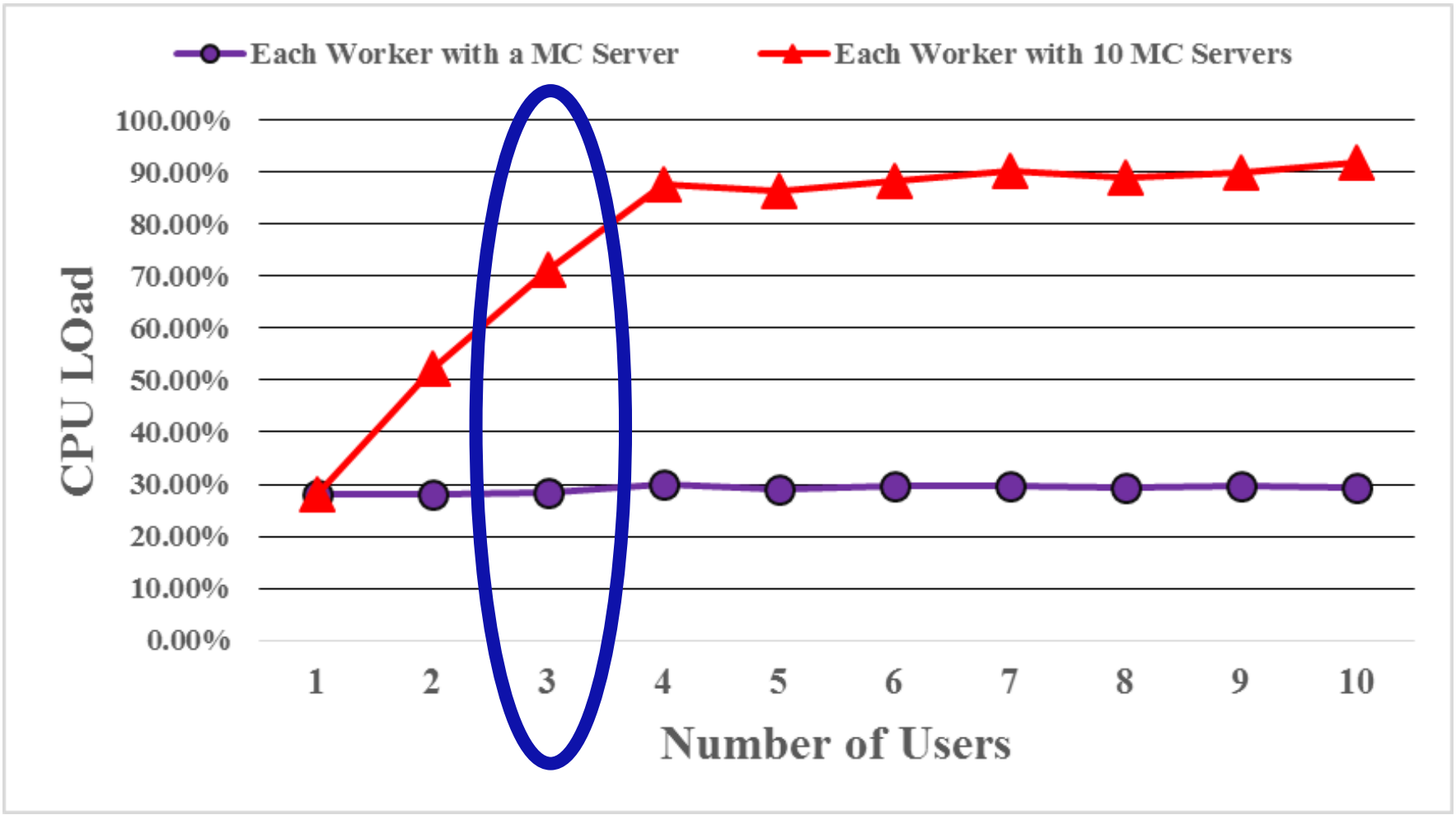
Wheel rims of automobiles

OS	Windows 7
CPU	2 vCPU
RAM	4 GB

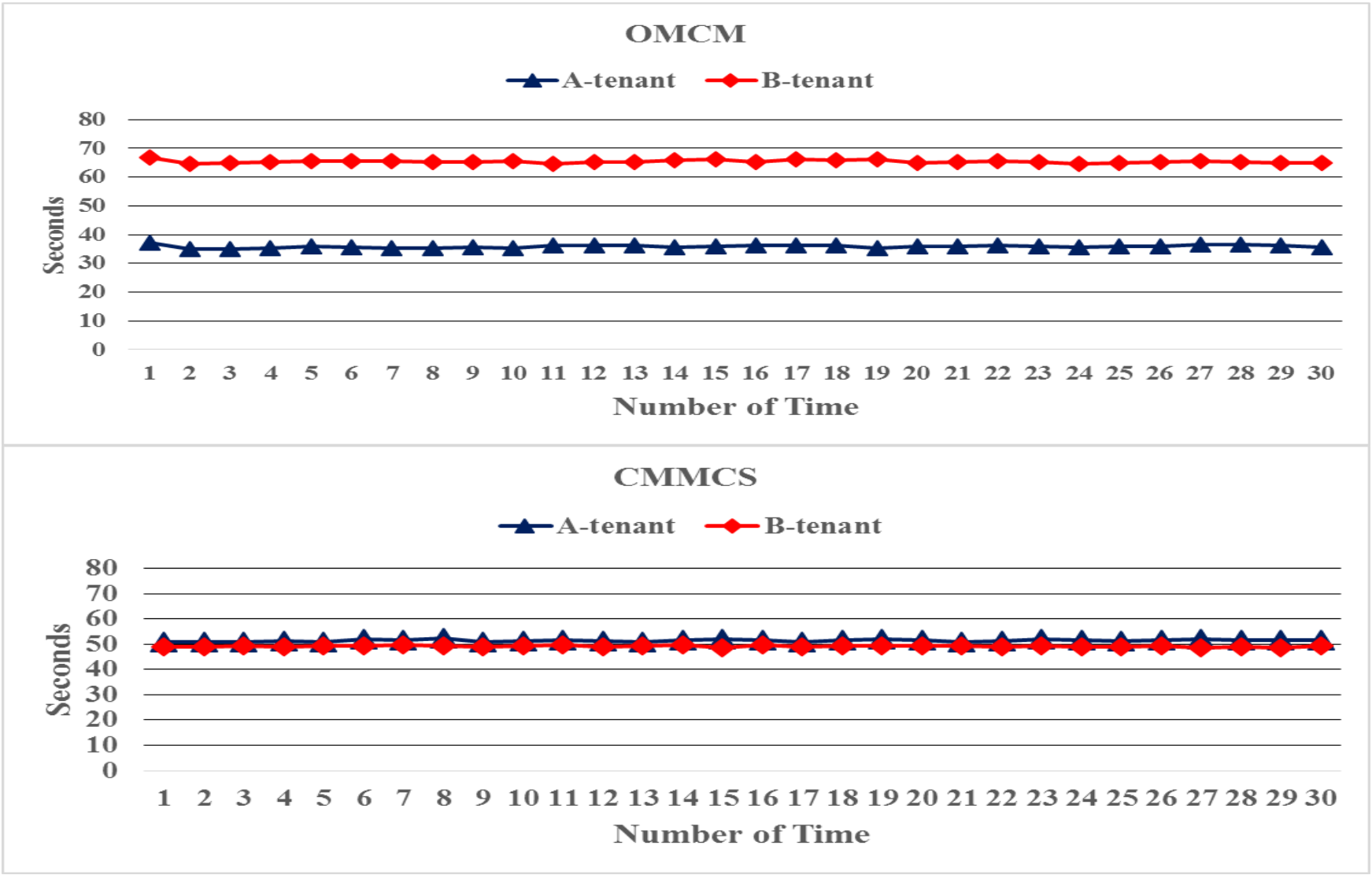
Comparison of the model creation time



Comparison of the usage rate of the virtual machine



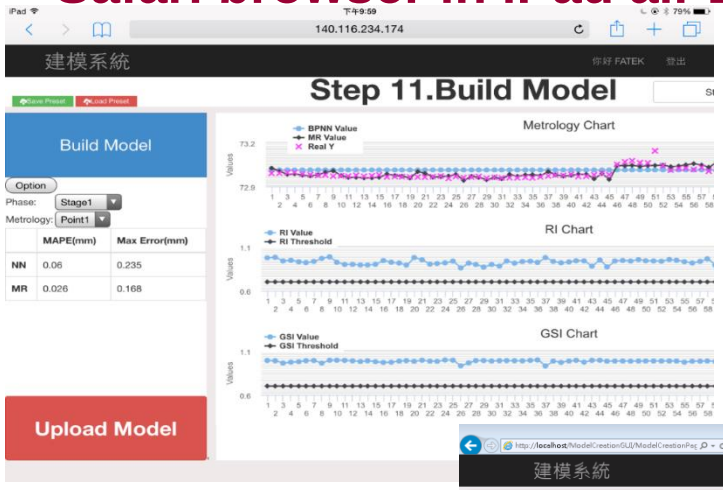
Comparison of the efficacy of the CMMCS and OMCM in a multi-tenant model creation scenario



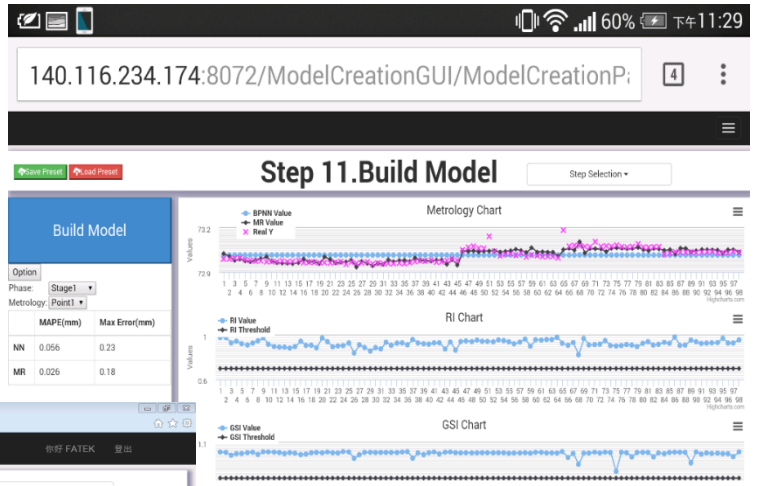
HTML 5-based GUIs

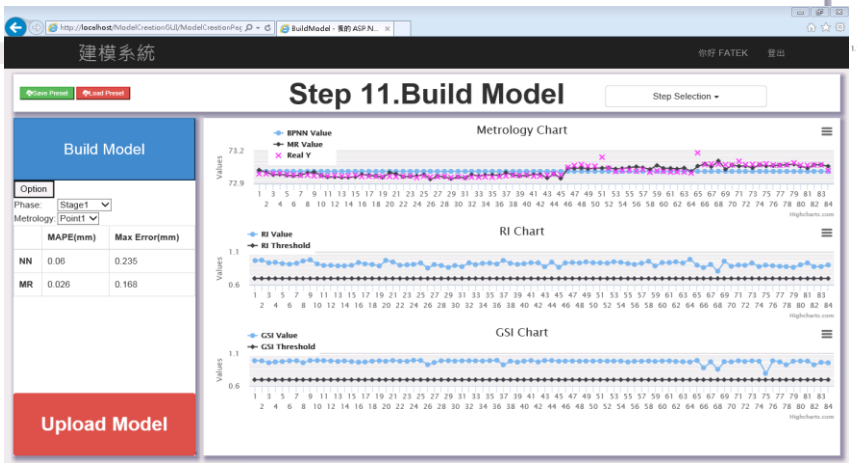
- We also download the HTML 5-based GUIs of the CMMCS to the browsers in Windows 7, iPad Air 2, and Android.

Safari browser in iPad air 2



Chrome browser in Android





IE browser in windows 7

VI. 雲製造服務自動建置機制
MSACS
(Manufacturing Services Automated Construction Scheme)

- 榮獲2018 IEEE機器人與自動化國際研討會(ICRA 2018)最佳自動化論文決賽入圍。(2018/5)
- 刊登於IEEE Robotics and Automation Letter (RA-L)國際期刊： (2018/7)

C.-C. Chen, M.-H. Hung*, P.-Y. Li, Y.-C. Lin, Y-Y. Liu, and F.-T. Cheng, "A Novel Automated Construction Scheme for Efficiently Developing Cloud Manufacturing Services," IEEE Robotics and Automation Letters (RA-L), Vol. 3, No. 3, pp. 1378-1385, July 2018.

智慧製造平台
先進製造物聯雲AMCoT

- 榮獲2018 IEEE自動化科學與工程國際研討會(CASE 2107)最佳應用論文獎。(2017/8)
- 刊登於IEEE Robotics and Automation Letter (RA-L)國際期刊： (2017/7)

Y.-C. Lin, M.-H. Hung*, H.-C. Huang, C.-C. Chen, H.-C Yang, Y.-S. Hsieh, and F.-T. Cheng, "Development of Advanced Manufacturing Cloud of Things (AMCoT) - A Smart Manufacturing Platform," *IEEE Robotics and Automation Letters (RA-L)*, Vol. 2, No. 3, pp. 1809-1816, July 2017.

Cloud Manufacturing Paradigm

- Cloud Manufacturing (CMfg):
  - CMfg has become a hot research topic for Smart Manufacturing.
  - CMfg's main concept is to encapsulate and virtualize distributed manufacturing resources (both hardware and software) into cloud services that can be used on-demand over networks to support manufacturing activities.

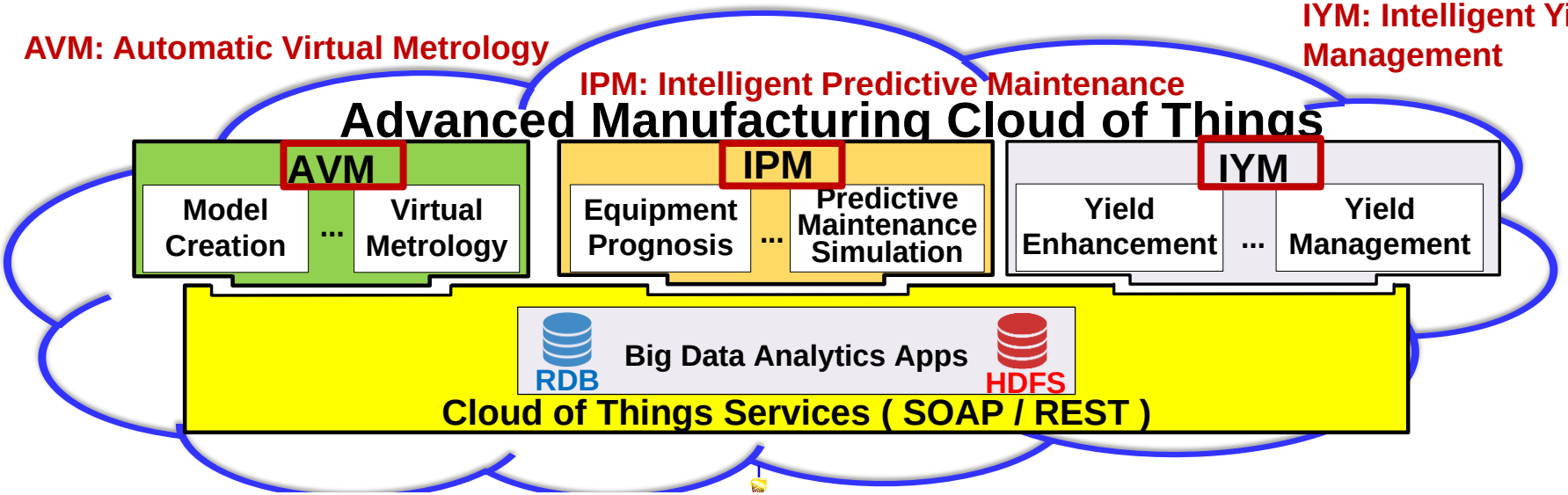
Advanced Manufacturing Cloud of Things

- Cloud Computing: abundant resources; potential latency
- Edge Computing: limited resources; more responsive

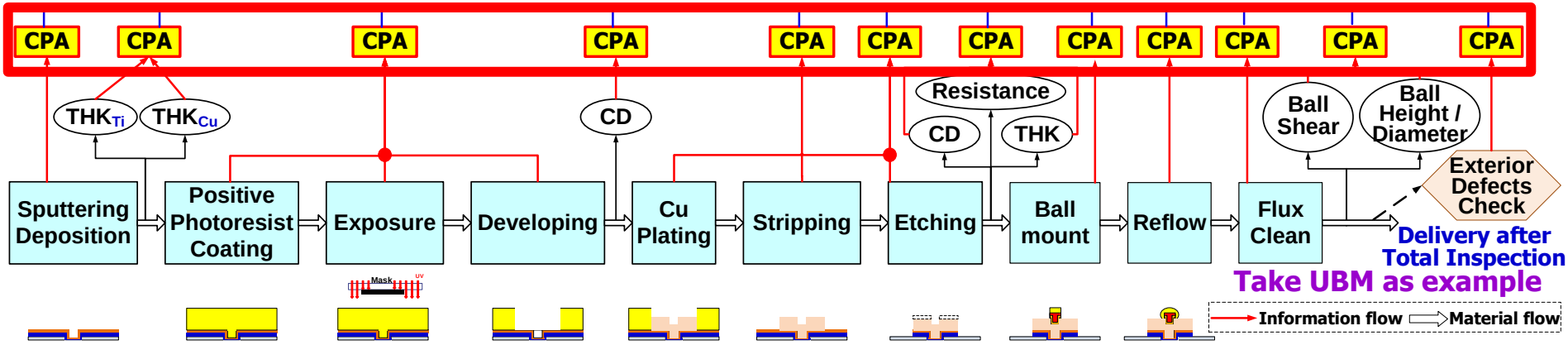
AVM: Automatic Virtual Metrology

IPM: Intelligent Predictive Maintenance

IYM: Intelligent Yield Management

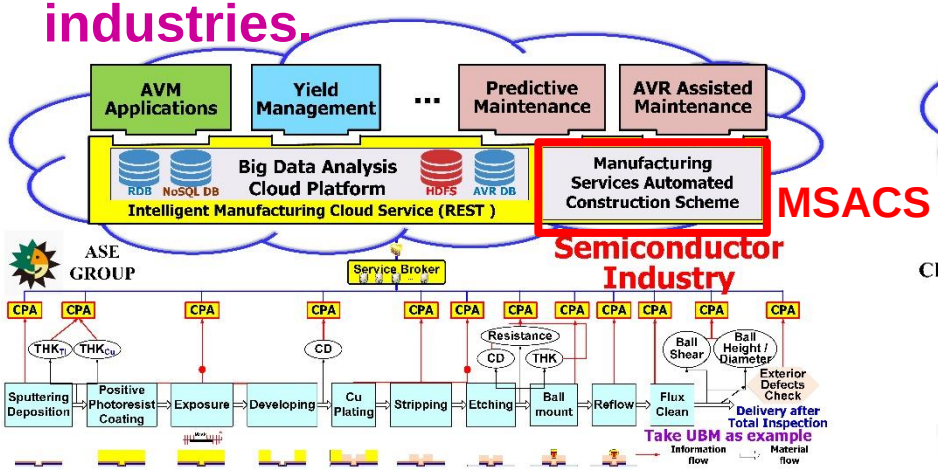


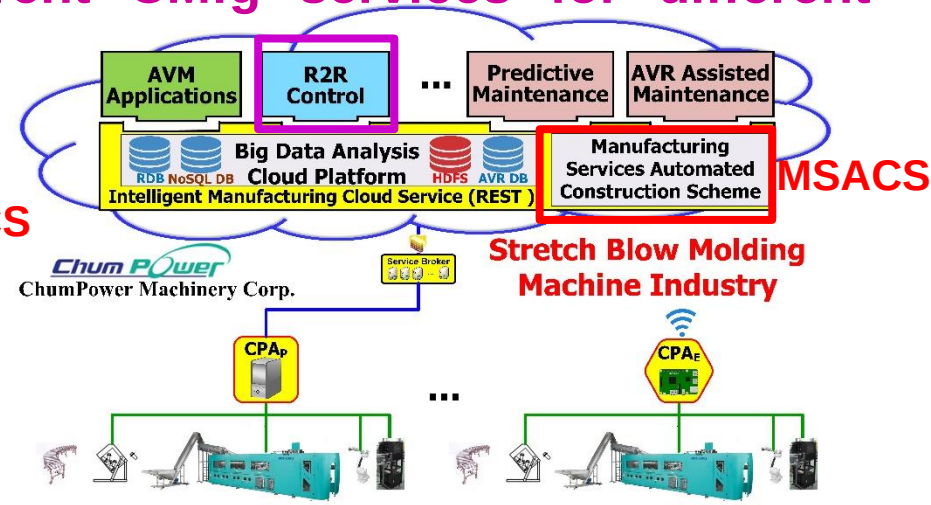
- CPA (Cyber-Physical Agent): an intelligent manufacturing edge device

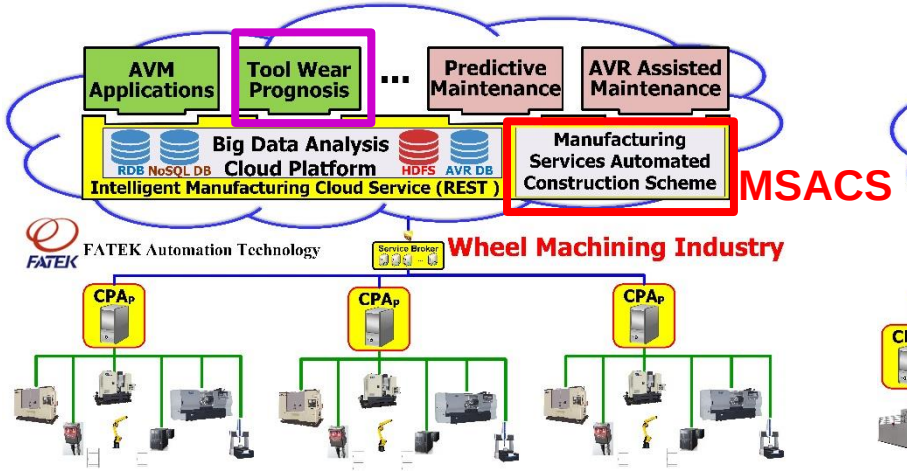


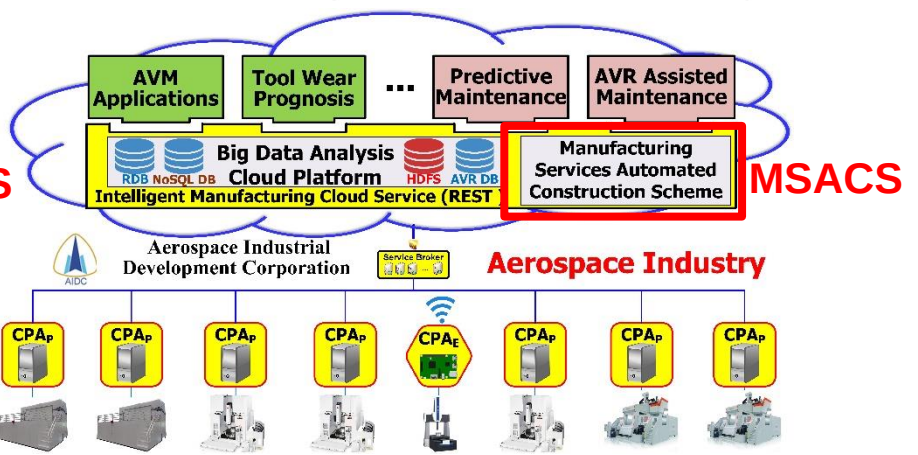
AMCoT's Industrial Application Scenarios

- AMCoT has been successfully applied to several manufacturing industries
- AMCoT needs to provide different CMfg services for different industries.









Challenges of constructing cloud manufacturing services

- **How to constructing cloud manufacturing services automatically and effectively** is **a necessary and challenging issue in promoting cloud manufacturing services**.
- In the service computing field, there are many service composition methods which can optimally **compose multiple atomic web services to create new value-added services**.
- However, these **previous service composition-related works** assumed that there were some existing services to work on and **did not address how to build them automatically**.
- **Manually constructing CMfg services using standalone software library packages (SSLP) involves many efforts and would become complicated and time-consuming.**

Objective of MSACS

- Aimed at facilitating rapid construction of CMfg services, we proposes an automated construction scheme of CMfg services, called **MSACS (Manufacturing Service Automated Construction Scheme)**, by adopting **a text-based-template and automated-code-generation approach** and **leveraging technologies of JSON, RESTful Web Service, and Command Scripting**.
- In industrial case studies, we apply MSACS to build the **AVM cloud service** and the **IYM cloud service** on the intelligent manufacturing platform AMCoT.
- Testing results demonstrate that **MSACS can automatically construct the target CMfg services in a very efficient manner** after uploading the required Standalone Software Library Packages (SSLP).

製造服務自動建置機置
Manufacturing Service Automated Construction Scheme (MSACS)

IEEE

IEEE ROBOTICS AND AUTOMATION SOCIETY

Best Paper Award in Automation—Finalist

IEEE International Conference on Robotics and Automation – ICRA 2018

21-25 May 2018 in Brisbane, Australia

is hereby presented to

Fan-Tien Cheng

For the paper entitled

"A Novel Automated Construction Scheme for Efficiently Developing Cloud Manufacturing Services"

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. FOUNDED NEW YORK 1884

24 May 2018

Alex Zelinsky
General Co-Chair, ICRA 2018

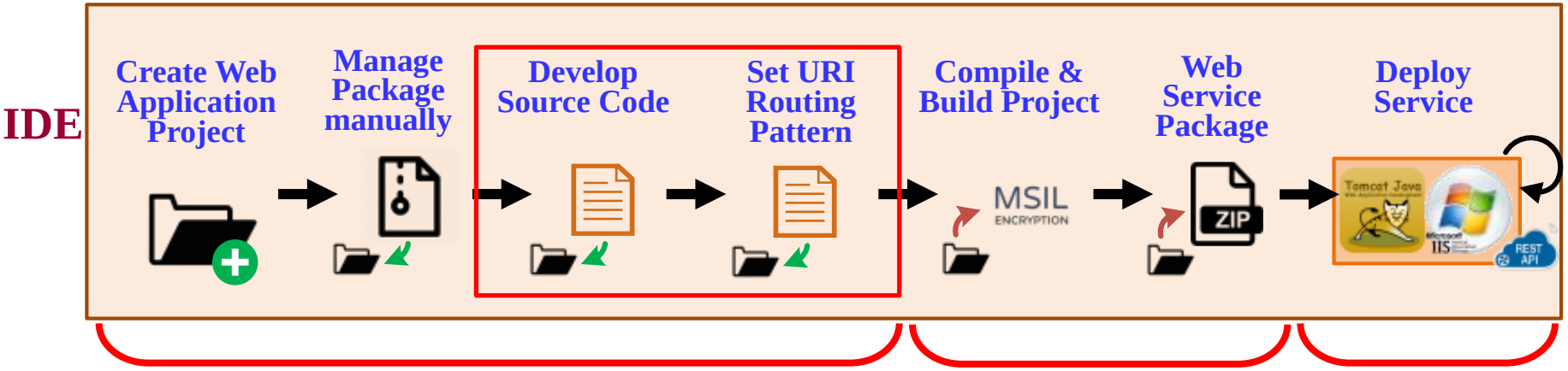
Nancy Amato
Awards Co-Chair, ICRA 2018

Frank Park
General Co-Chair, ICRA 2018

Greg D. Hager
Awards Co-Chair, ICRA 2018

Procedure of Manually Developing a CMfg Service

- Develop a CMfg service manually using IDE (Integrated Development Environment):



- Example Code of a Web Method

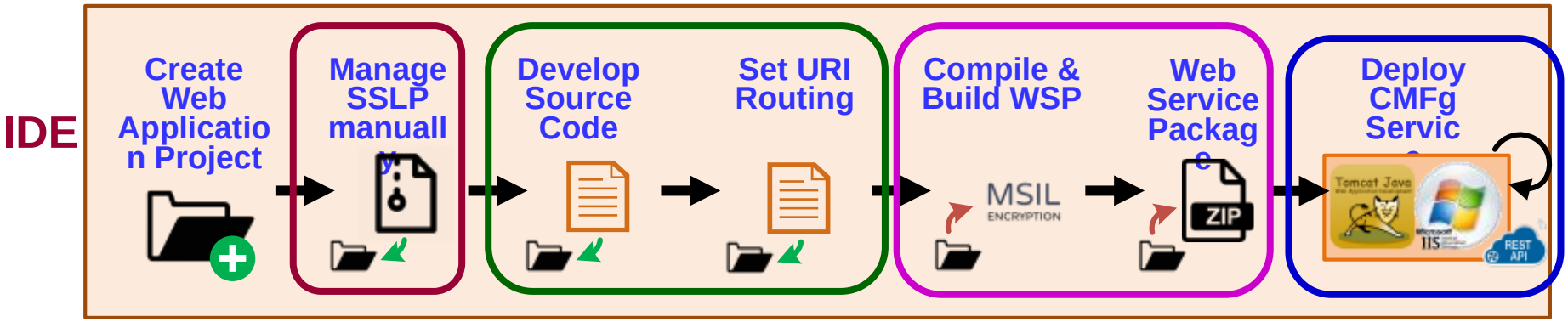
```
using Newtonsoft.Json.Linq;
using AVMService_WebApi.Model;
using AVMService;

namespace AVMService_WebApi.Controllers
{
    public class ServiceController : ApiController
    {
        [HttpPost]
        public string ModelFanOut([FromBody]Parameters parameters)
        {
            try{
                Service obj = new Service();
                string[] Data = JSONDecoders.DecodeJSONArray(parameters.postData);
                string input1 = Data[0];
                string input2 = Data[1];
                string input3 = Data[2];
                string input4 = Data[3];
                String result;
                result = obj.ModelFanOut(input1, input2, input3, input4);
                return result.ToString();
            }
            catch(Exception e){
                return e.ToString();
            }
        }
```

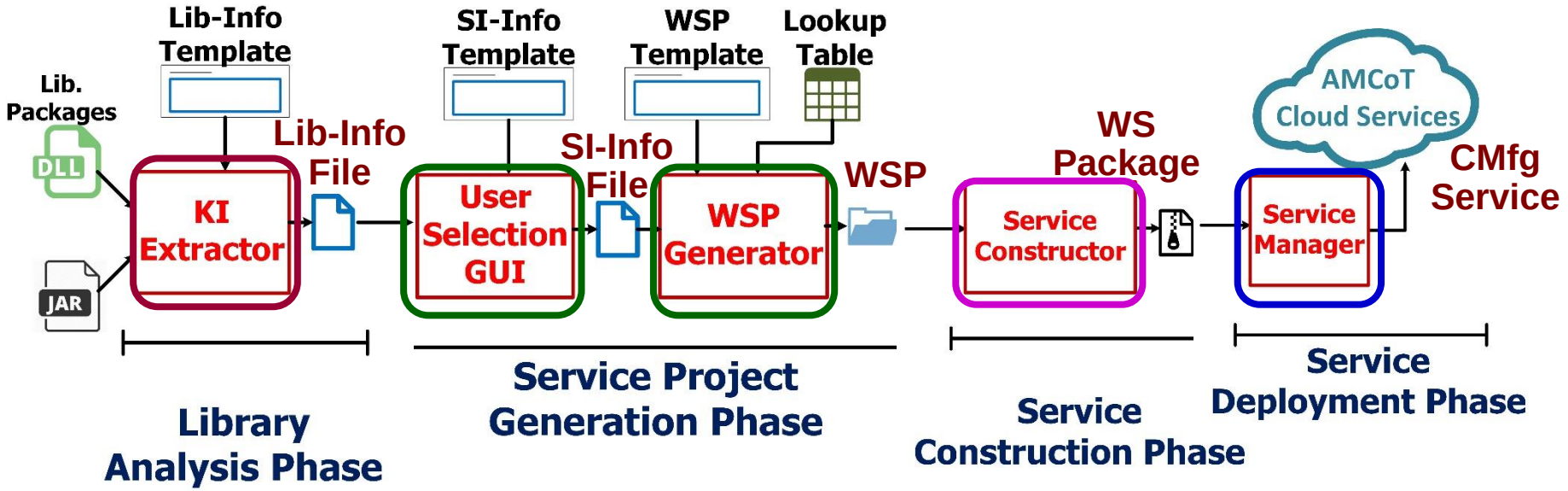
Need to repeat n times of such coding if there are n Web methods, which is time-consuming and error-prone.

Workflow Design of MSACS

- **Manual Workflow of Building a CMfg service:**

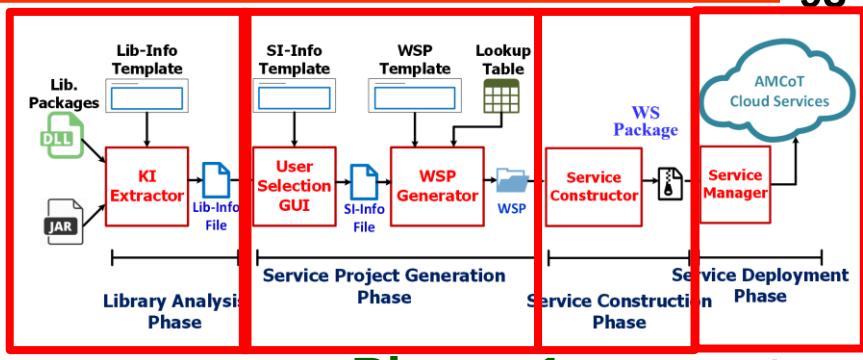


- **Automated Workflow of MSACS for building a CMfg service :**



Architecture and Operational Flows of MSACS

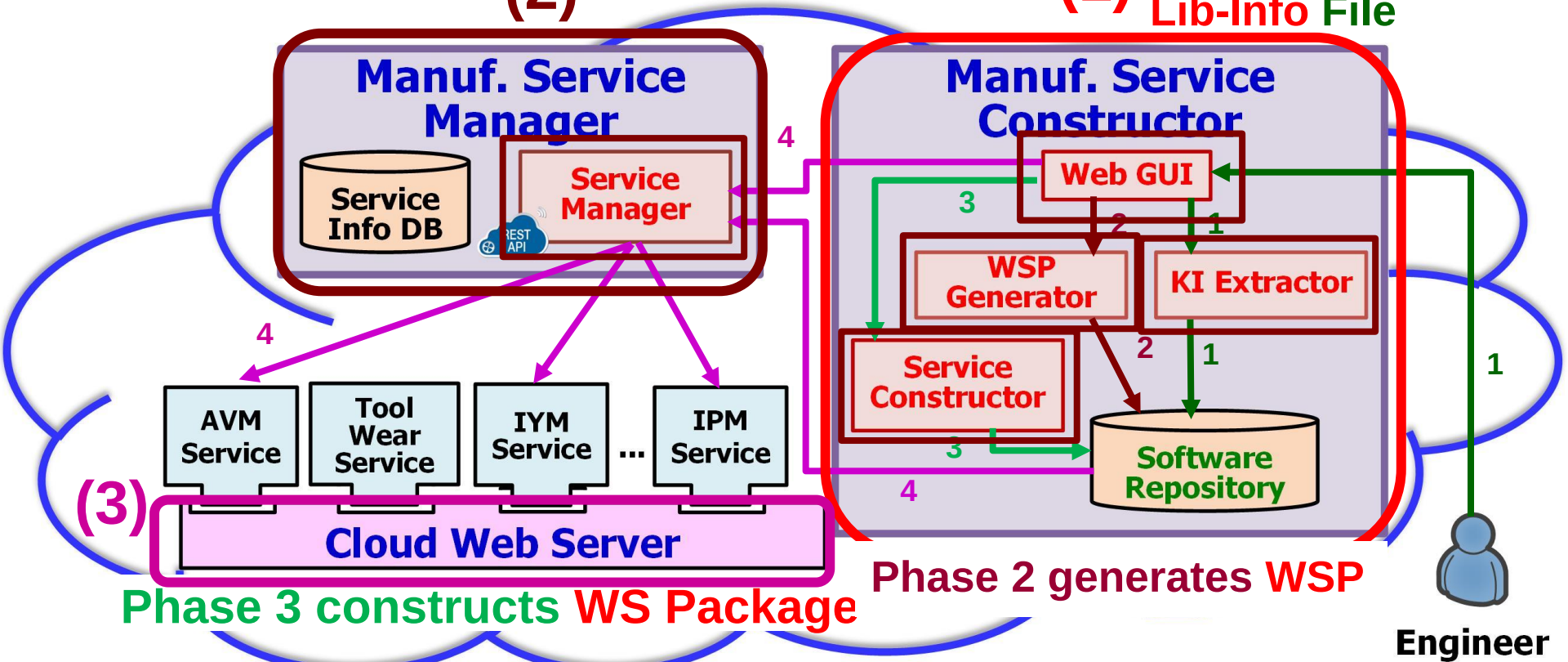
- MSACS is a cloud-based system whose architecture consists of three parts.



(1) Phase 1 generates Lib-Info File

Phase 4 deploys CMfg Service

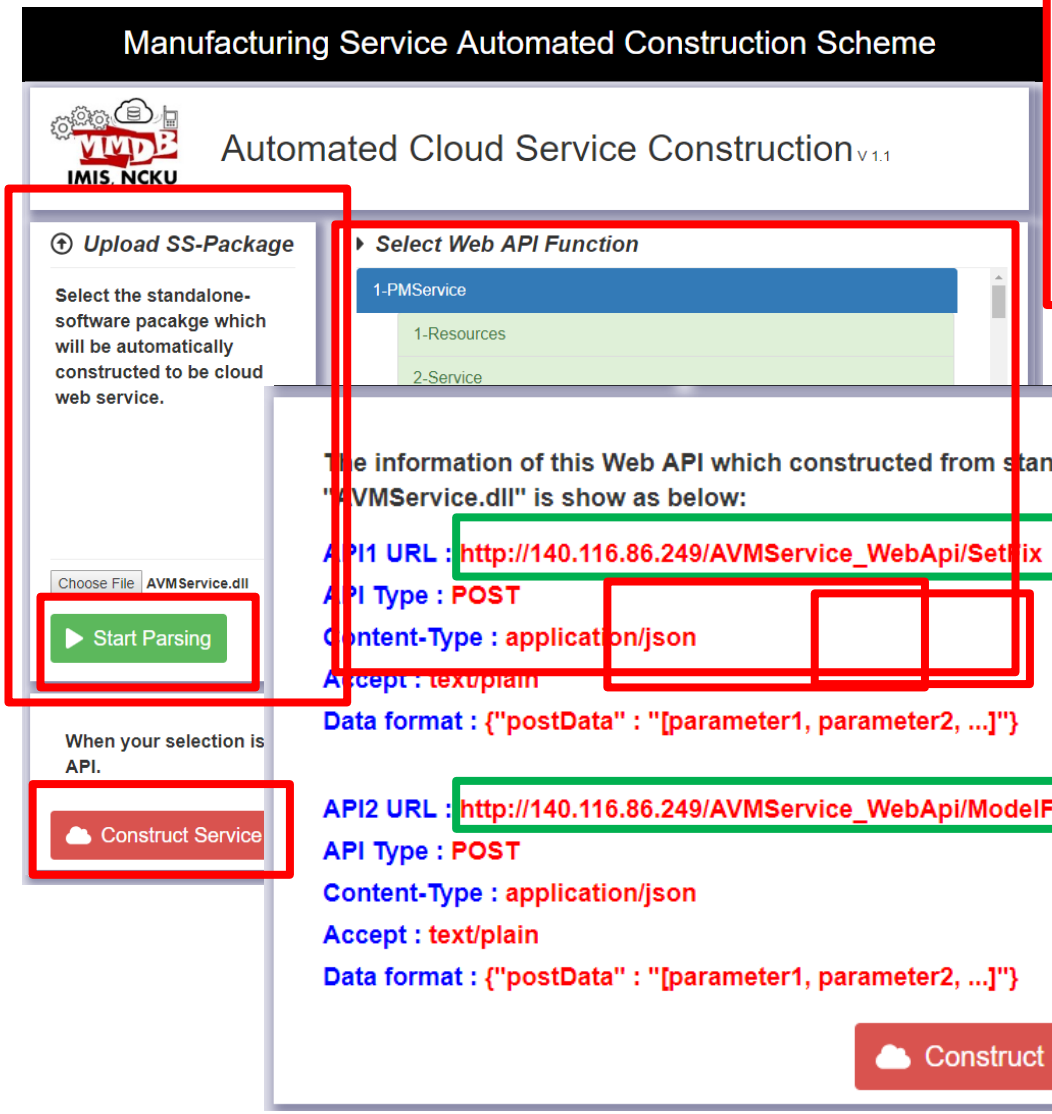
(2)

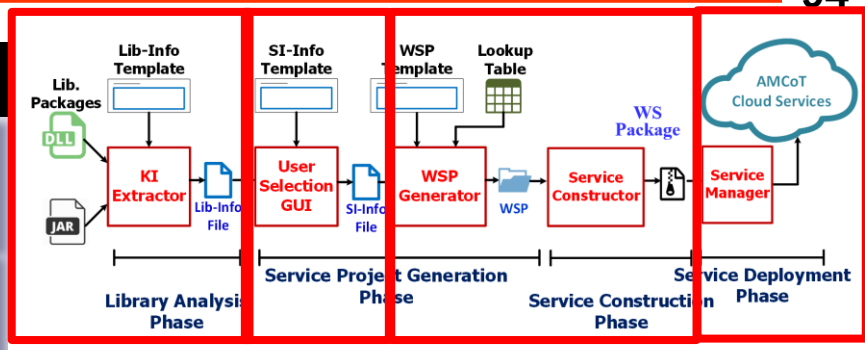


(3) Phase 3 constructs WS Package

Phase 2 generates WSP

Illustration of MSACS' Operations with Web GUIs

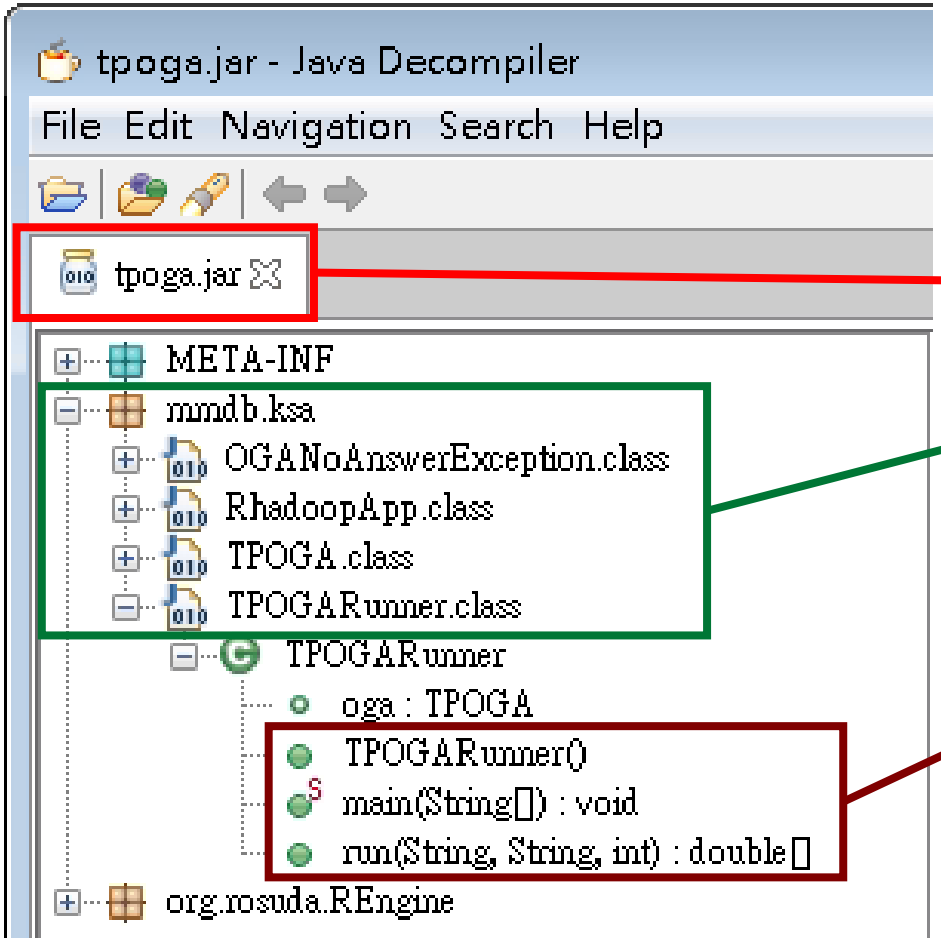




Lib-Info File | SI-Info File | WSP | WS Package | CMfg Service

Three-Layer Hierarchical Information of Java Jar Library

- **Structure of a Java Jar Library in a decompiler:**



1. Jar SSLP's Name

JarInputStream Object ($JISO$)

Jar Name

JarEntry Object (JEO_i)

Package Name

Class Name List

2. Package Names and Class Names

Function Content (FC_j)

Function Name

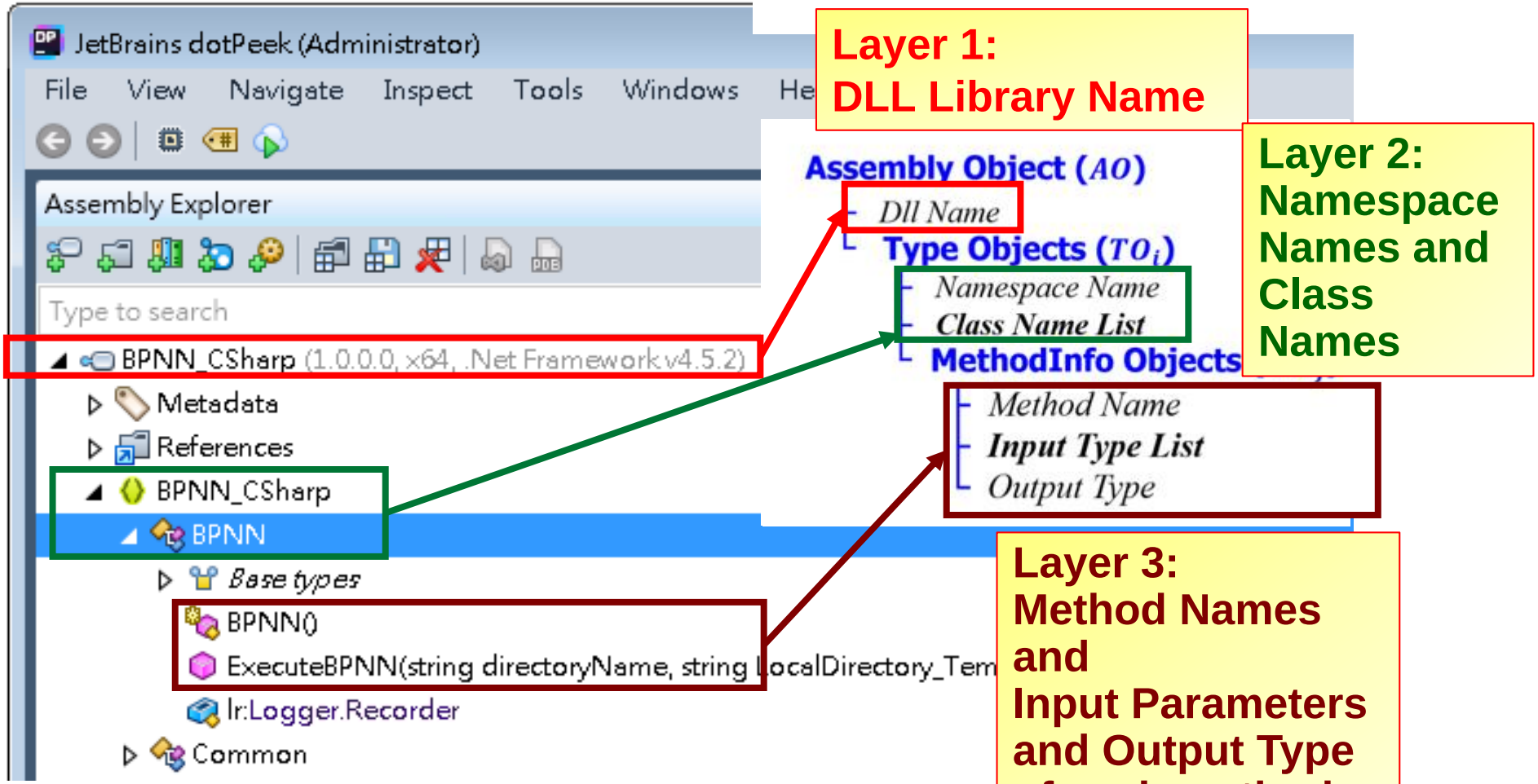
Input Type List

Output Type

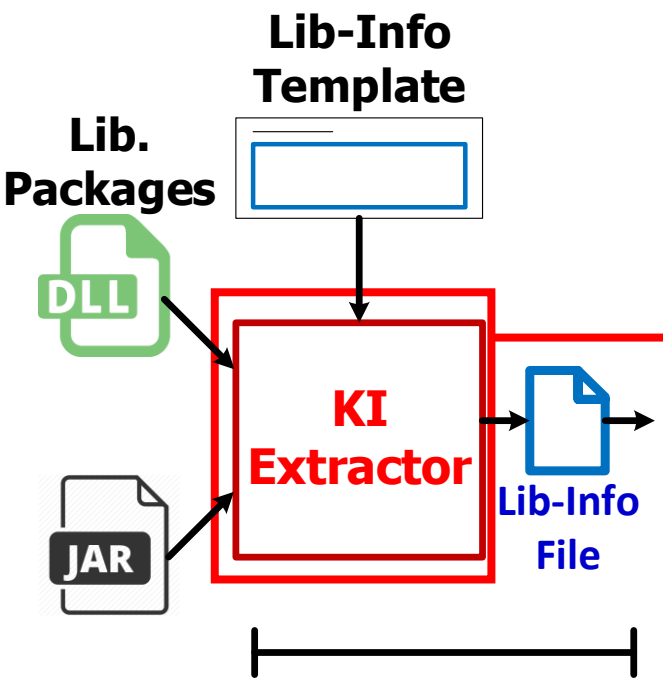
3. Method Names and Input Parameters & Output Type of each method

Three-Layer Hierarchical Information of C# DLL Library

- **Structure of a C# DLL Library in a decompiler:**

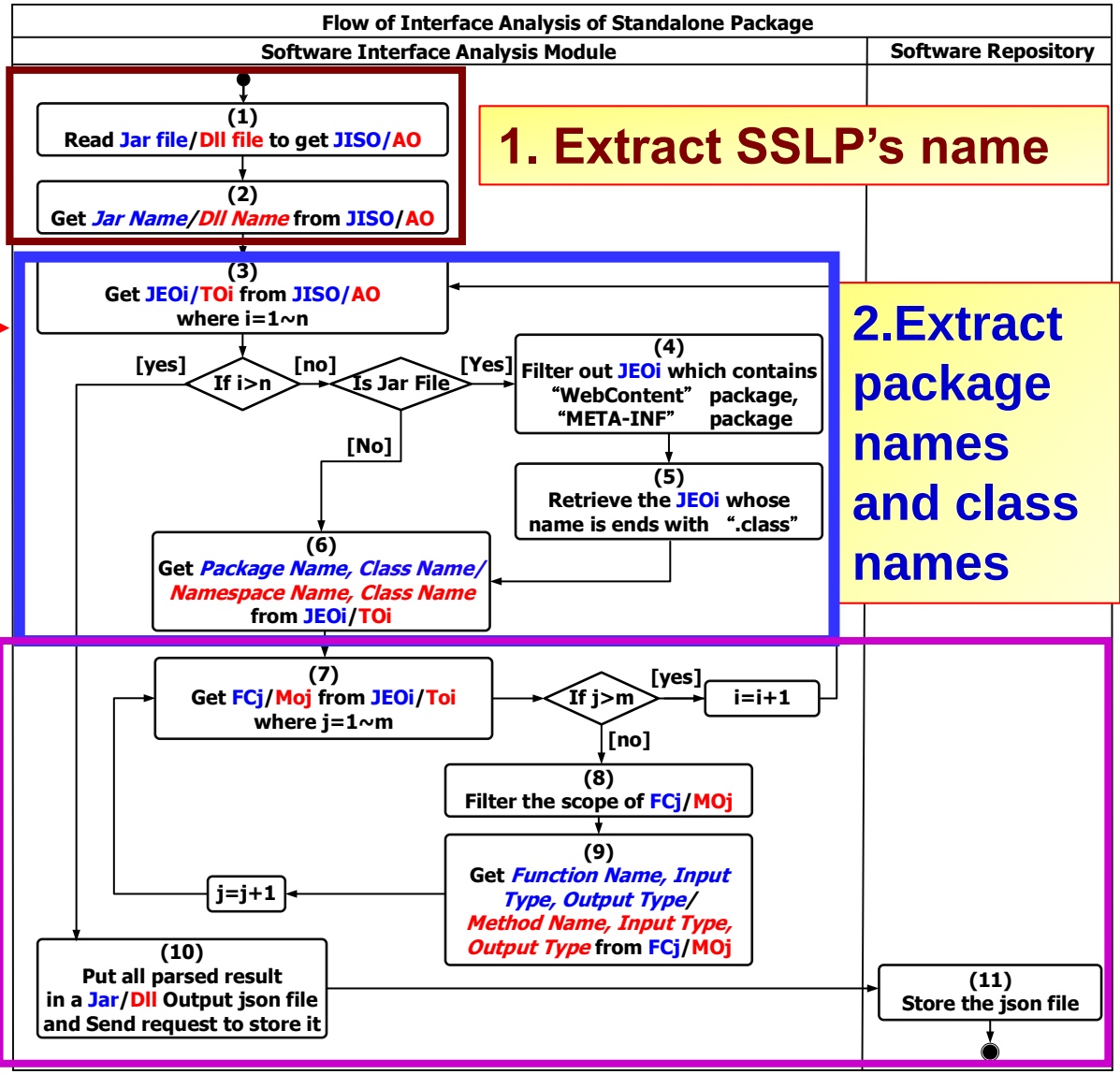


Generic Algorithm of KI Extractor for both C# DLL and Java Jar Libraries



Library Analysis Phase

3. Extract method names, and the input parameters and output type of each method



1. Extract SSLP's name

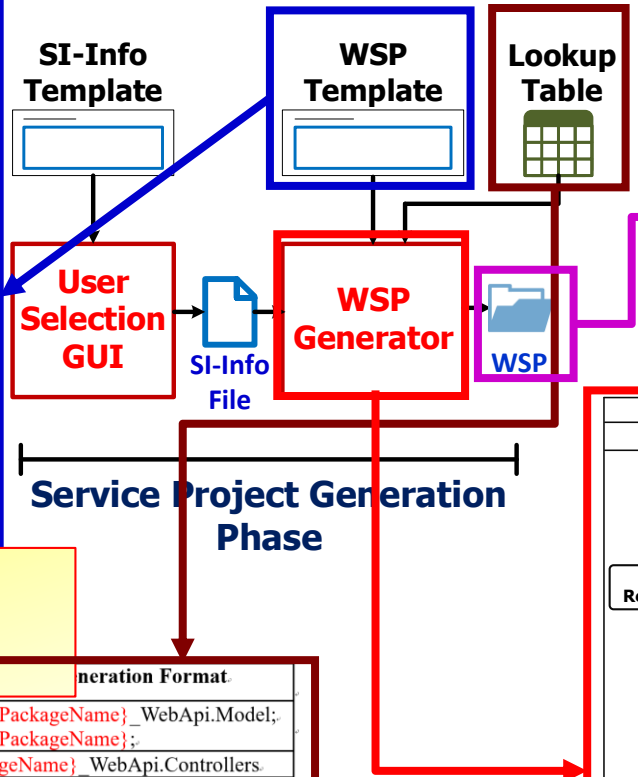
2.Extract package names and class names

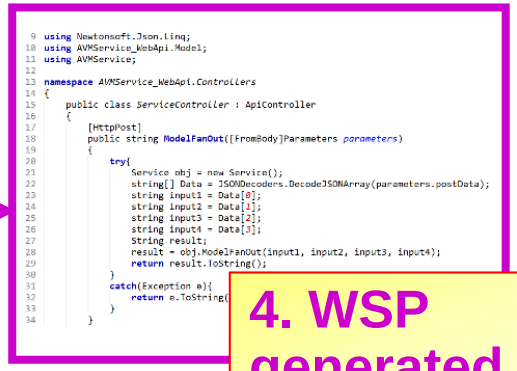
Functional Modules of WSP Generator


```
using Newtonsoft.Json.Linq;
//@WSP:ImportDllFile

namespace //@WSP:NamespaceDefinition
{
    public class //@WSP:ClassDefinition
    {
        //@WSP:HttpType
        public string //@WSP:FunctionName
        {
            try{
                //@WSP:ClassDefinitionForSSLP
                //@WSP:InputTypeDefinition
                //@WSP:OutputTypeDefinition
                //@WSP:ExecutionMethodForSSLP
                //@WSP:SuccessReturn
            }
            catch(Exception e){
                return e.ToString();
            }
        }
```

1. WSP Template

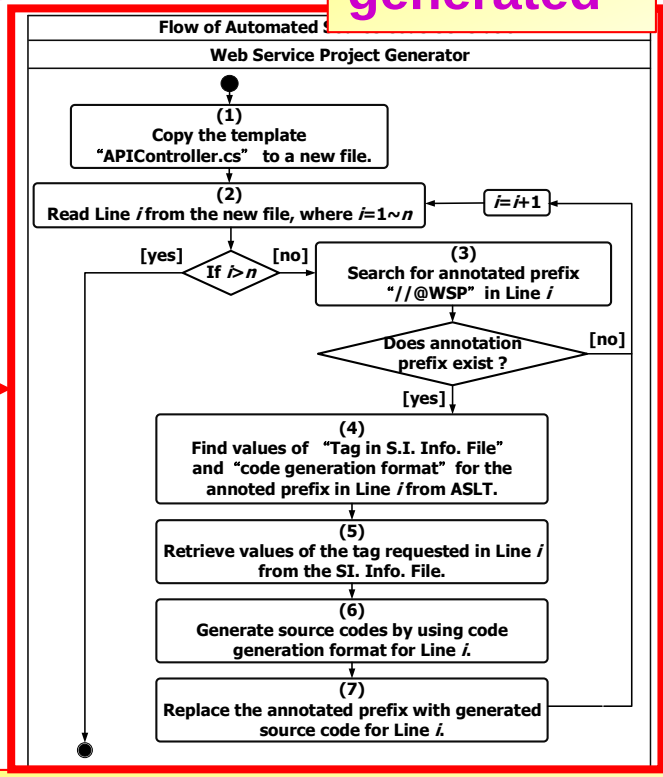




4. WSP generated

Annotated Key		neration Format
//@WSP:ImportDllFile	PackageName	using {PackageName}_WebApi.Model; using {PackageName};
//@WSP:NamespaceDefinition	PackageName	{PackageName}_WebApi.Controllers
//@WSP:ClassDefinition	ClassName	{ClassName}Controller:ApiController
//@WSP:HttpType	APIType	[Http{APIType}]
//@WSP:FunctionName	MethodName	{MethodName}([FromBody]Parameters parameters)
//@WSP:ClassDefinitionForSSLP	ClassName	{ClassName} obj = new {ClassName}();
//@WSP:InputTypeDefinition	InputType	String[] Data = JSONDecoders; DecoderJsonArray(paremeters.postData); {InputType} input1 = Data[0]; {InputType} input2 = Data[1];
//@WSP:OutputTypeDefinition	OutputType	{OutputType} result;
//@WSP:ExcutionMethodForSSLP	MethodName	rsult=obj.{MethodName}(input1,input2);
//@WSP:SuccessReturn	-	Return result.ToString();

2. Lookup Table



3. Code-Generation Algorithm

Performance Evaluation of MSACS

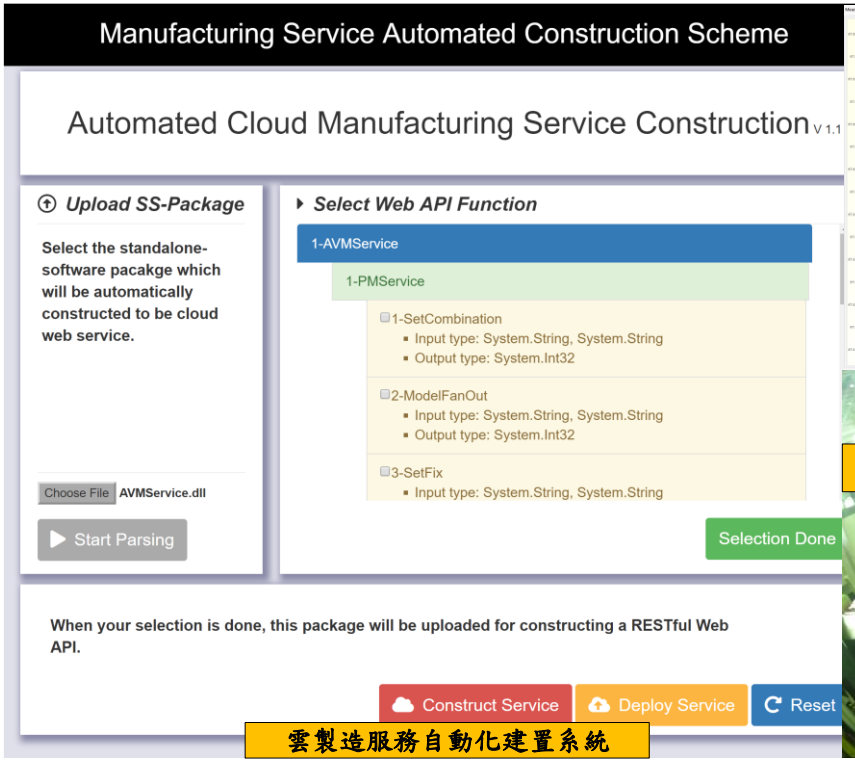
- Performance comparisons of a Novice, a Skilled Programming Professional, and MSACS in creating CMfg services:

Experiments	Novice	Skilled Programing Professional	MSACS
(1) Construction of an AVM Web API using a C# SSLP (The ExecuteBPNN method in the BPNN_CSharp.dll library)	23 hr. 2 min	10.88 min	0.4 min
(2) Construction of an IYM Web API using a Java SSLP (The run method in the tpoga.jar library)	30 hr. 30 min	33.18 min	1.12 min

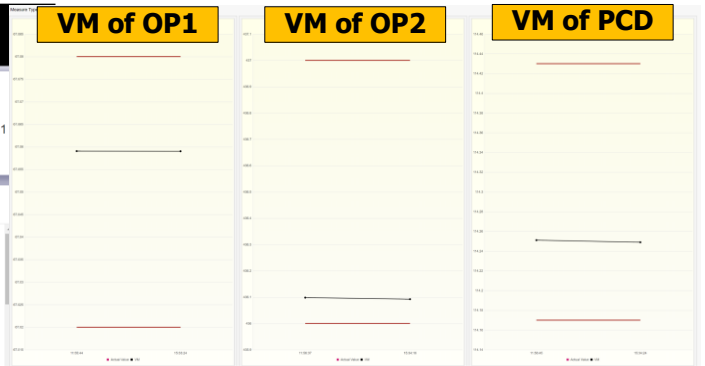
➔ MSACS is 30 times on average quicker than the professional programmer (22 min vs 0.76 min) and more than 1600 times on average quicker than the novice.

DEMO of Scenario

Construct AVM Service using MSACS and Conduct on-line Virtual Metrology for Wheel Machining

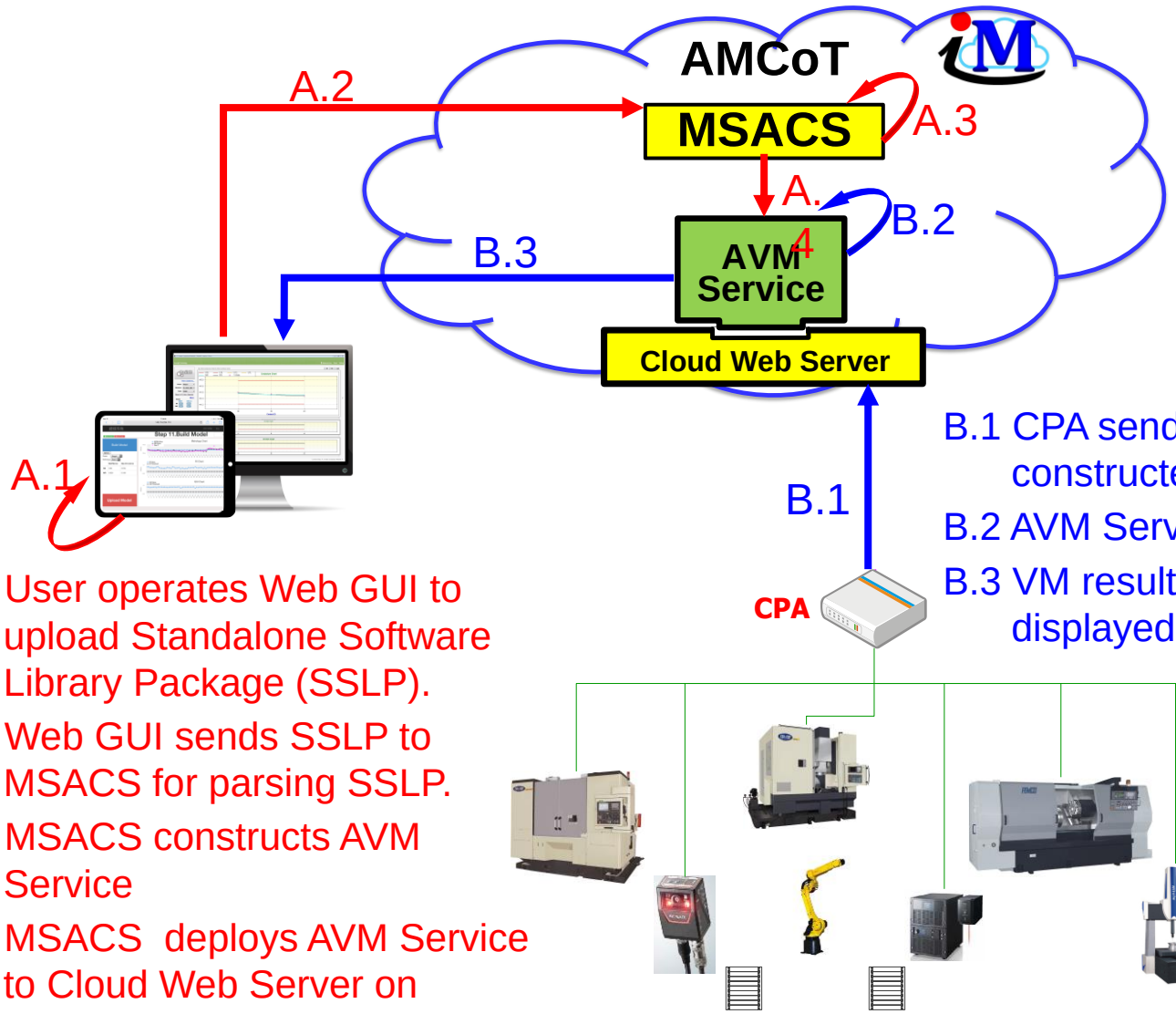


雲製造服務自動化建置系統



鋁輪圈加工虛擬量測應用

Scenario 1: Construct AVM Service using MSACS on AMCoT and Conduct on-line Virtual Metrology for Wheel Machining



A.1 User operates Web GUI to upload Standalone Software Library Package (SSLP).

A.2 Web GUI sends SSLP to MSACS for parsing SSLP.

A.3 MSACS constructs AVM Service

A.4 MSACS deploys AVM Service to Cloud Web Server on AMCoT

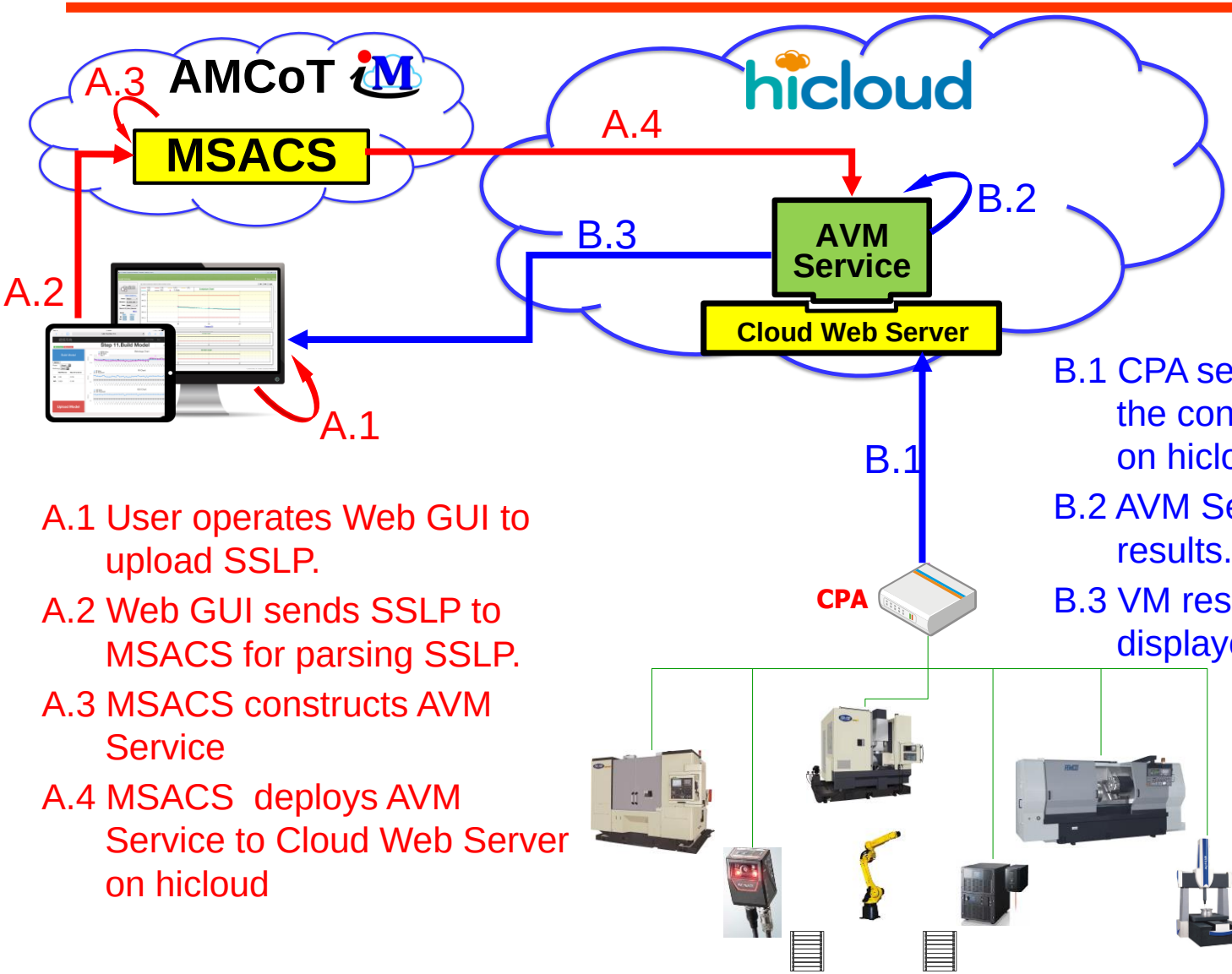
B.1 CPA sends process data to the constructed AVM Service on AMCoT.

B.2 AVM Service computes VM results.

B.3 VM results are sent to and displayed on Web GUI.

Automatic Deployment of Automatic Virtual Metrology (AVM) Services to the private cloud of iMRC

Scenario 2: Construct AVM Service using MSACS on hicloud and Conduct on-line Virtual Metrology for Wheel Machining



A.1 User operates Web GUI to upload SSLP.

A.2 Web GUI sends SSLP to MSACS for parsing SSLP.

A.3 MSACS constructs AVM Service

A.4 MSACS deploys AVM Service to Cloud Web Server on hicloud

B.1 CPA sends process data to the constructed AVM Service on hicloud.

B.2 AVM Service computes VM results.

B.3 VM results are sent to and displayed on Web GUI.

Automatic Deployment of AVM Services to hicloud