PAAS AND SAAS

DR. JUNE SUNG PARK, KAIST

http://flavors.me/june_sung_park

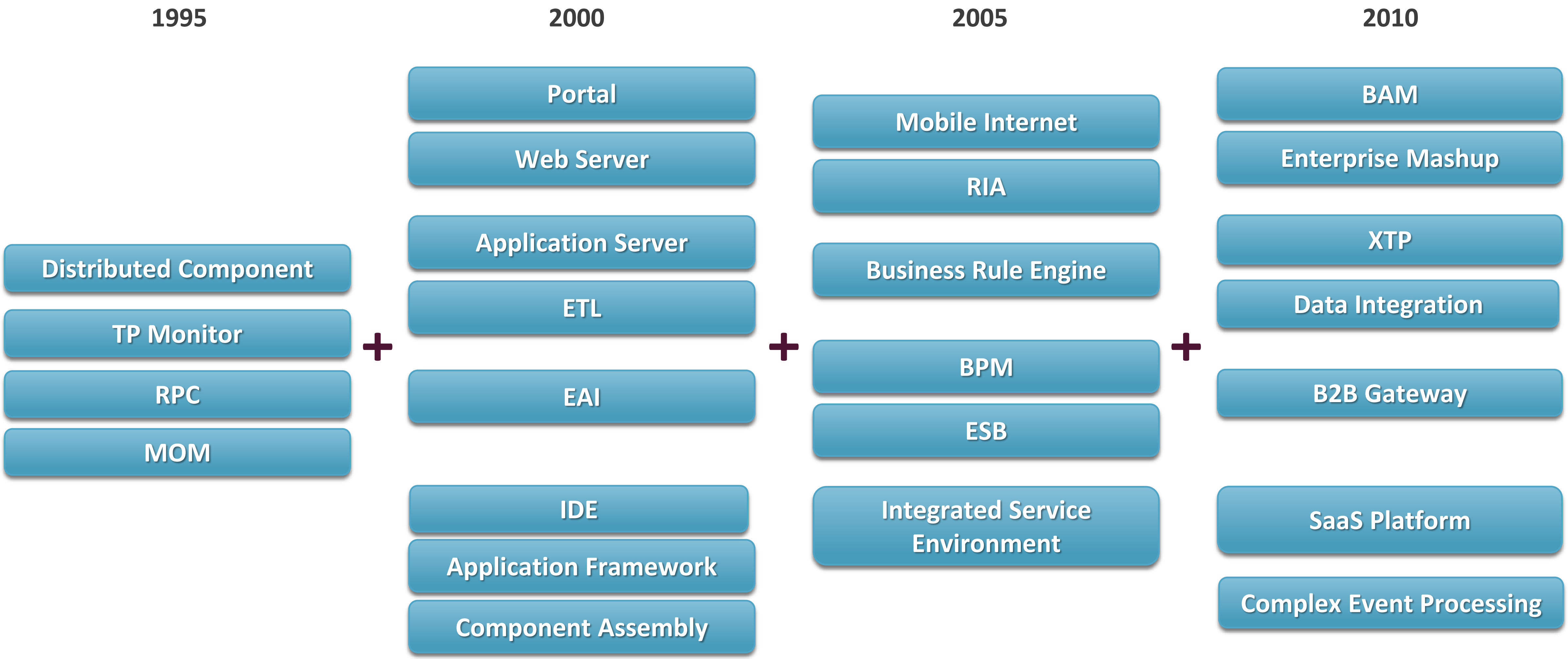
MODULE OBJECTIVE

- Understand the evolution of middleware towards PaaS.
- Understand required functionalities of PaaS.
- Understand PaaS architecture.
- Understand the taxonomy of PaaS.
- Understand SaaS development platforms.
- Understand the maturity model of SaaS.
- Understand difference among ASP, On-Demand Packaged Apps and SaaS service models.
- Understand SaaS architecture.

DEFINITION OF PAAS

- Middleware is a key technology layer in business and consumer computing. It underlies, enables and controls most modern applications, and is of equal importance in cloud computing.
- To achieve the technical benefits of cloud computing—for example, self-service management and elastic scaling— middleware must be cloud-enabled.
- Platform as a service (PaaS) is cloud-enabled middleware offered as a subscription service with diverse capabilities, including application platforms, database management systems (DBMSs), mobile back-end services, integration platforms, business process management (BPM) and rule engines, business analytics platforms, complex-event processing (CEP), event-streaming services and in-memory computing (IMC) platforms.
- Cloud-enabled middleware may also be offered as software distribution. PaaS software is typically used by IT organizations to create a private PaaS, but could also be used by software vendors to create a public PaaS or SaaS offering of their own.
- Finally, custom-made, cloud-enabled middleware may be embedded in SaaS and not directly exposed to the subscribers.

EVOLUTION OF MIDDLEWARE



1995

- Distributed Component
- TP Monitor
- RPC
- MOM

+

2000

- Portal
- Web Server
- Application Server
- ETL
- EAI
- IDE
- Application Framework
- Component Assembly

+

2005

- Mobile Internet
- RIA
- Business Rule Engine
- BPM
- ESB
- Integrated Service Environment

+

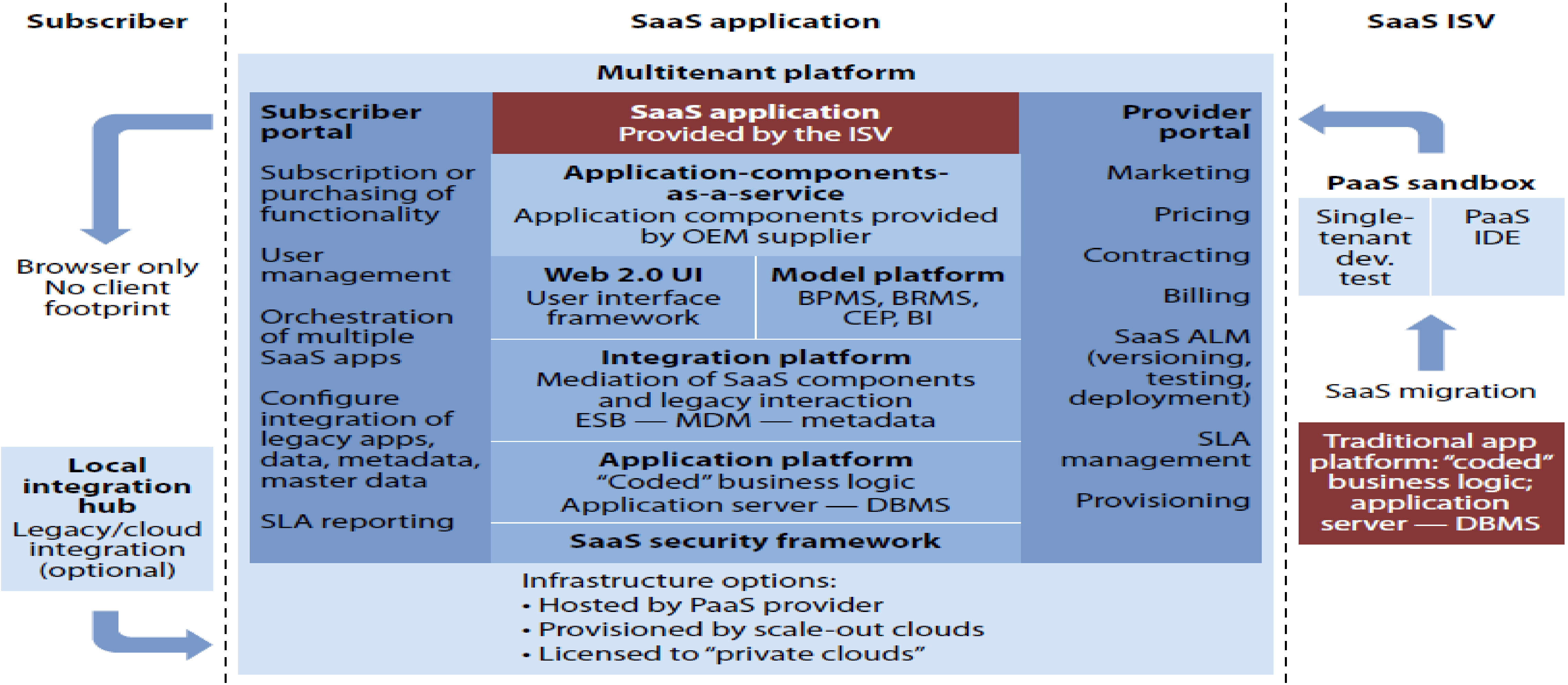
2010

- BAM
- Enterprise Mashup
- XTP
- Data Integration
- B2B Gateway
- SaaS Platform
- Complex Event Processing

REQUIREMENTS FOR PAAS

- Elastic, on-demand allocation and de-allocation of shared resources (elastic scaling). This capability had better be automated.
- Isolation of logical tenants (i.e., instances of applications that typically correspond to distinct subscribing organizations) isolation in a physically shared context.
- Tenant-specific processing of application logic and data management.
- Tenant-specific provisioning, management, monitoring, high availability and disaster recovery procedures.
- Tenant-specific tracking of resource use, costing and billing.
- Tenant-specific security, privacy and integrity of data and operations.
- Tenant-specific configuration, customization and version control.
- Tenant-specific self-service administration.
- Nested multitenancy for tenants that are SaaS vendors and support their own tenants (subtenants)—including subtenant security, tracking, billing and version control.
- Advanced Web-scale performance characteristics, including extreme scalability, availability, response times and cost controls (typically facilitated through use of parallelization, standardization and IMC).
- Global, continuous and ubiquitous accessibility.

PAAS ARCHITECTURE



Subscriber

Browser only
No client footprint

Local integration hub
Legacy/cloud integration (optional)

SaaS application

Multitenant platform

Subscriber portal

Subscription or purchasing of functionality

User management

Orchestration of multiple SaaS apps

Configure integration of legacy apps, data, metadata, master data

SLA reporting

SaaS application
Provided by the ISV

Application-components-as-a-service
Application components provided by OEM supplier

Web 2.0 UI
User interface framework

Model platform
BPMS, BRMS, CEP, BI

Integration platform
Mediation of SaaS components and legacy interaction
ESB — MDM — metadata

Application platform
"Coded" business logic
Application server — DBMS

SaaS security framework

Provider portal

Marketing

Pricing

Contracting

Billing

SaaS ALM (versioning, testing, deployment)

SLA management

Provisioning

Infrastructure options:
- Hosted by PaaS provider
- Provisioned by scale-out clouds
- Licensed to "private clouds"

SaaS ISV

PaaS sandbox

Single-tenant dev. test

PaaS IDE

SaaS migration

Traditional app platform: "coded" business logic; application server — DBMS

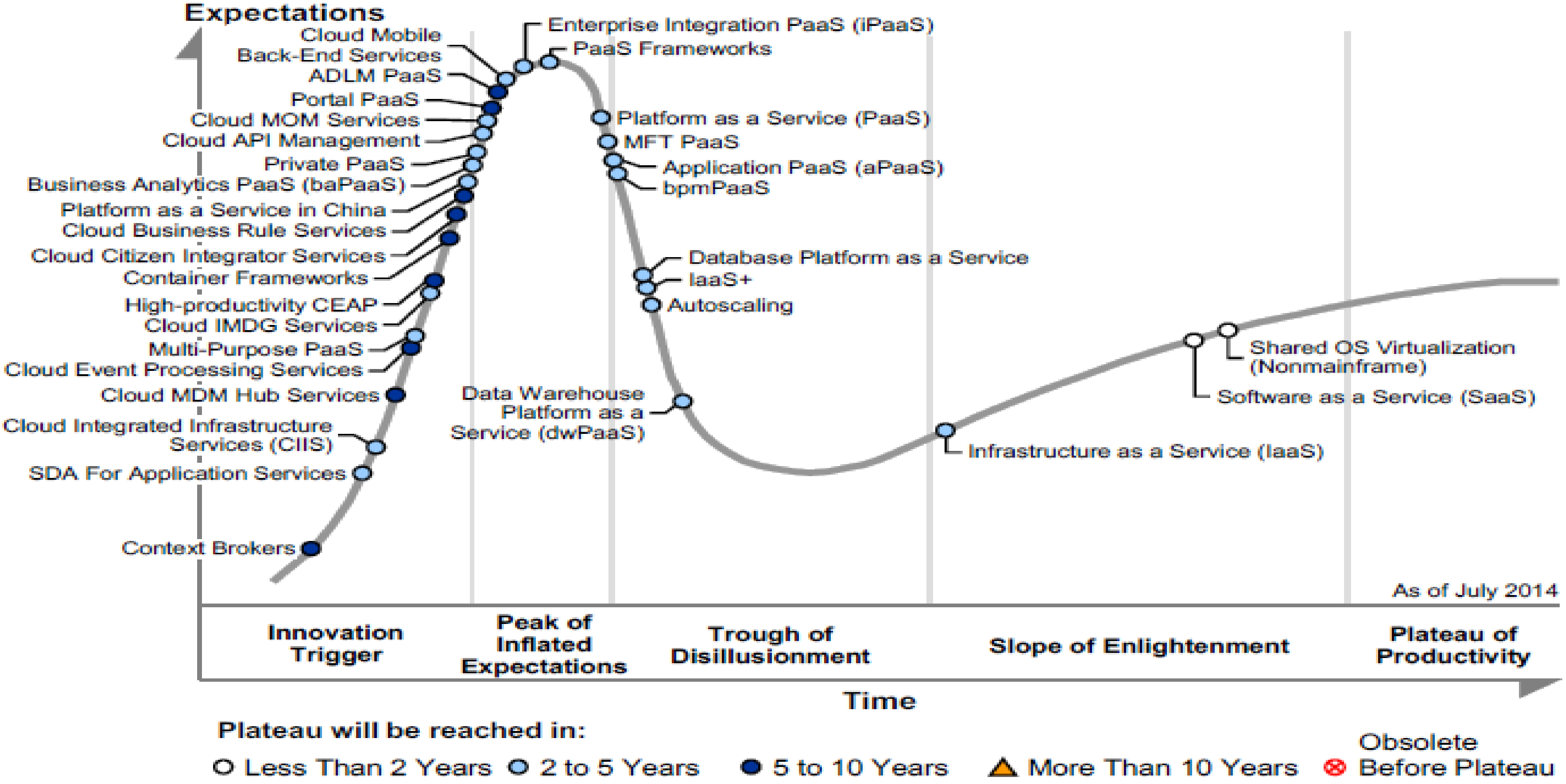
Forrester Research, The Forrester Wave™: Platform-as-a-Service for Vendor Strategy Professionals, Q2 2011, 2011.

PAAS TAXONOMY

- Development Platform Services (dPaaS)
- Application Platform Services (aPaaS)
- Integration Platform Services (iPaaS)
- Business Process Management Platform Services (bpmPaaS)
- Business Rule Platform Services
- DBMS and Data Store Services (dbPaaS)
- Application Development Life Cycle Management Services (ADLM PaaS)
- Business Analytics Services (baPaaS)
- Enterprise Horizontal Portal Services
- MDM (Meta Data Management) Hub Services
- Mobile Back-End Services (MBaaS)
- IMDG (In-Memory Data Grid) Services
- Event-Processing Services
- API Management Services
- MOM (Message-Oriented Middleware) Services
- MFT (Managed File Transfer) Services

Gartner, Platform as a Service: Definition, Taxonomy and Vendor Landscape, 2014

PAAS HYPE CYCLE



Gartner, Hype Cycle for Platform as a Service, 2014

SAAS PLATFORM (APAAS)

- While traditional app platforms are designed for on-premise, single-tenant with no provision for elasticity, SaaS platforms require off-premise, multitenancy and elasticity.
- A SaaS platform should fulfill the same functions as the normal application platform, and additionally specialized could-enabled features.
- A tenant should be able to customize the SaaS user interface, business logic, data schema in real time without affecting the functionality or availability of the SaaS for all other tenants. The SaaS code base should be patched or upgraded without breaking tenant-specific customizations.
- By 2014 more than 20% of midsize to large enterprises will use SaaS platforms, up from less than 3% in 2009.

Normal App Platform Function	Cloud-Enabling Function
Business process management	**Global-class access**
Service-oriented architecture	**Multitenancy**
Application framework	**Tenant-aware customization**
Application server	**Tenant-aware security, privacy and integrity**
Message-oriented middleware	**Extreme Transaction Processing**
Enterprise service bus	**IDE in the cloud**
B2B integration	
Integrated Development Environment	

APAAS TREND

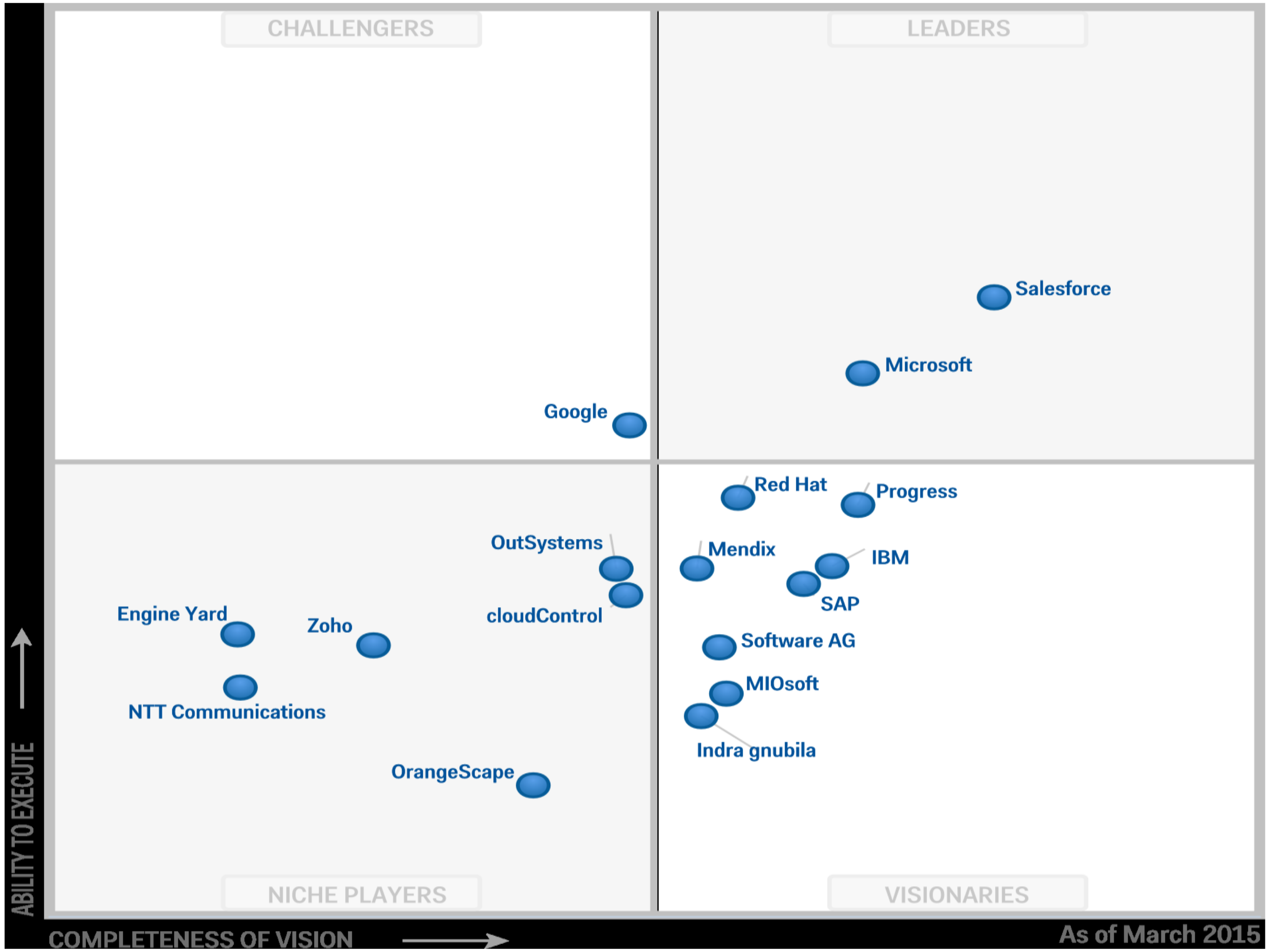
- Most platform technology innovation goes into the cloud platforms (the "cloud first" strategy). As the result, users seeking the leading-edge platform capabilities are pushed to cloud deployments.
- More on-premises application platforms are updated to be cloud-ready, drawing IT organizations to private aPaaS and then hybrid aPaaS.
- aPaaS offerings begin to challenge traditional platform middleware for mainstream projects, and not only for experimental, cloud-centric initiatives.
- The aPaaS market remains under construction — new versions of Cloud Foundry, Azure Cloud Services, IBM aPaaS and other offerings carry substantial internal design changes.
- PaaS frameworks (like Cloud Foundry, OpenShift and Apache Stratos) form the foundation for comprehensive PaaS suite offerings.
- New forms of xPaaS continue to emerge, driven by emerging use cases (e.g., business analytics, stream processing and mobile back end; IoT is expected to emerge as well) — most joining the core aPaaS capabilities to begin to form comprehensive PaaS offerings.
- Traditional middleware vendors, pushed by open-source and cloud-native PaaS competitors, enter the aPaaS market to defend their client base and boost their stagnant revenue, but face challenging business model transition issues..

Gartner, Magic Quadrant for Enterprise Application Platform as a Service, Worldwide, March 2015.

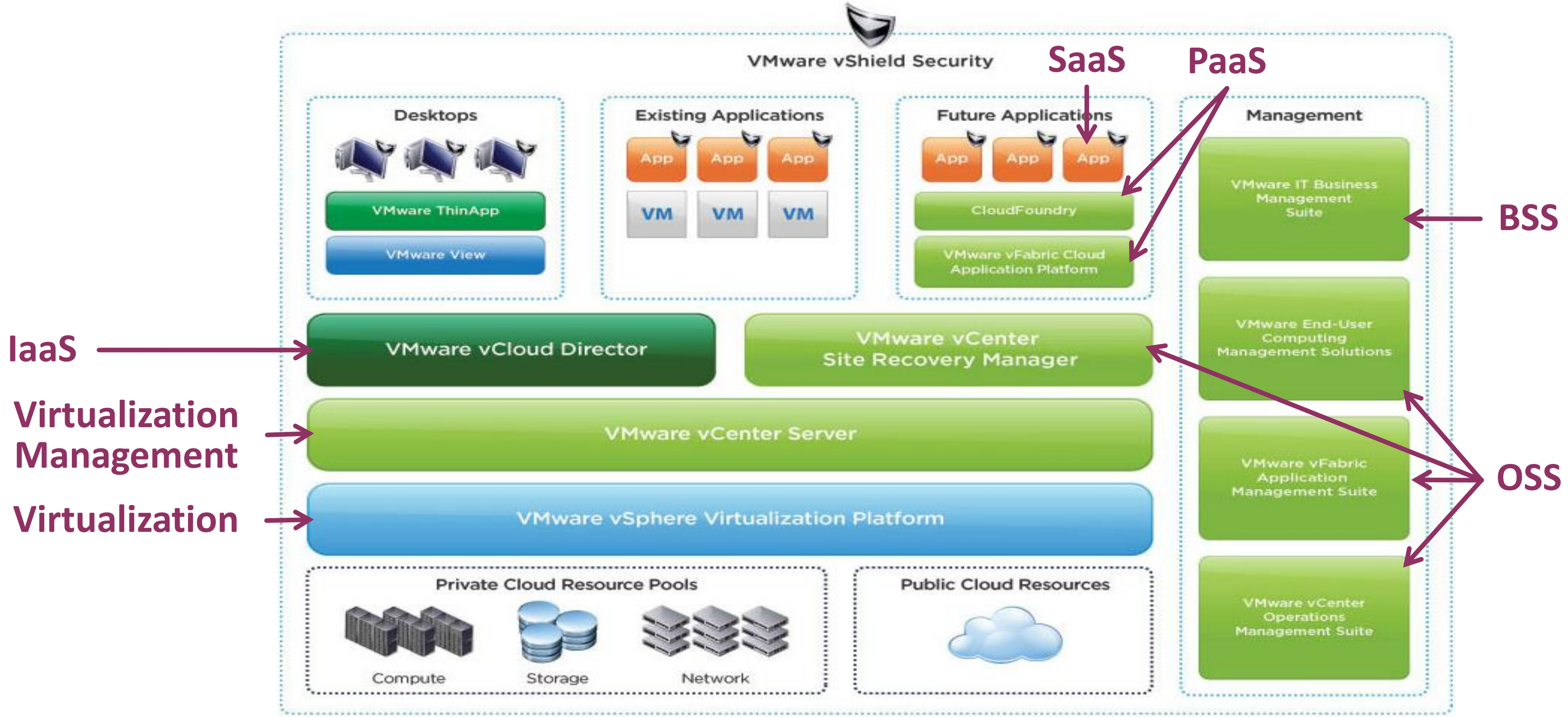
APAAS TREND

- Hybrid aPaaS emerges as the strategic platform for large organizations seeking to preserve their investment while embracing innovation; while medium and some small enterprises strategically focus on public aPaaS to gain the benefit of scale and excellence available in the public cloud that is unattainable with their limited resources.
- aPaaS leaders invest in offering both high-control and high-productivity services to meet the growing prevalence of bimodal IT.
- Open source has emerged as the new de facto standard — Linux, OpenStack, Cloud Foundry, Docker, Kubernetes, polyglot 3GL frameworks (Rails, Tomcat, etc.), MySQL, Hadoop, etc. All are seen by users as ensuring portability and reducing vendor lock-in.
- The divide between IaaS and aPaaS is becoming less pronounced as some leading platforms, like Microsoft Azure and Google Cloud Platform, offer a contiguous suite of IaaS and PaaS capabilities.
- The divide between aPaaS and SaaS is narrowing as well, as some aPaaS providers offer prebuilt business logic frameworks or libraries with their development environments to improve developer productivity, and in the process advance their aPaaS to aPaaS+.
- IoT pushes stream processing, event processing, real-time analytics, in-memory computing and real-time context-aware business decisions to the center of new applications and business solutions, putting pressure on platform providers (including aPaaS) to adjust and expand into digital business.

APAAS VENDORS



Gartner, Magic Quadrant for Enterprise Application Platform as a Service, Worldwide, March 2015

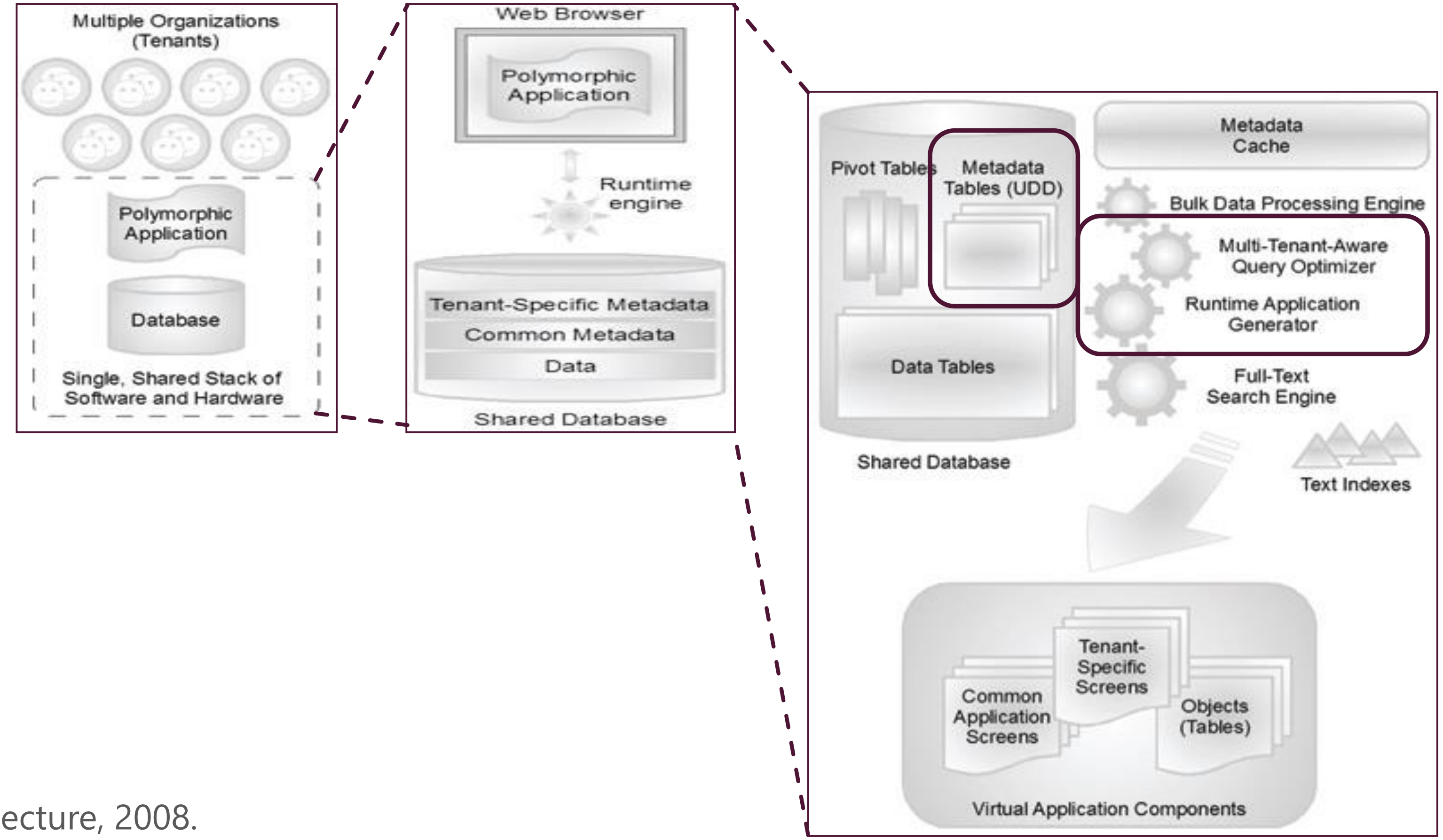

VMware vShield Security
SaaS
PaaS
Desktops
VMware ThinApp
VMware View
Existing Applications
App
App
App
VM
VM
VM
Future Applications
App
App
App
CloudFoundry
VMware vFabric Cloud Application Platform
Management
VMware IT Business Management Suite
BSS
IaaS
VMware vCloud Director
VMware vCenter Site Recovery Manager
VMware End-User Computing Management Solutions
Virtualization Management
VMware vCenter Server
VMware vFabric Application Management Suite
OSS
Virtualization
VMware vSphere Virtualization Platform
Private Cloud Resource Pools
Compute
Storage
Network
Public Cloud Resources
VMware vCenter Operations Management Suite

CASE STUDY: SAAS PLATFORM

Cloud Foundry

Force.com supports two ways to create custom applications and their individual components:

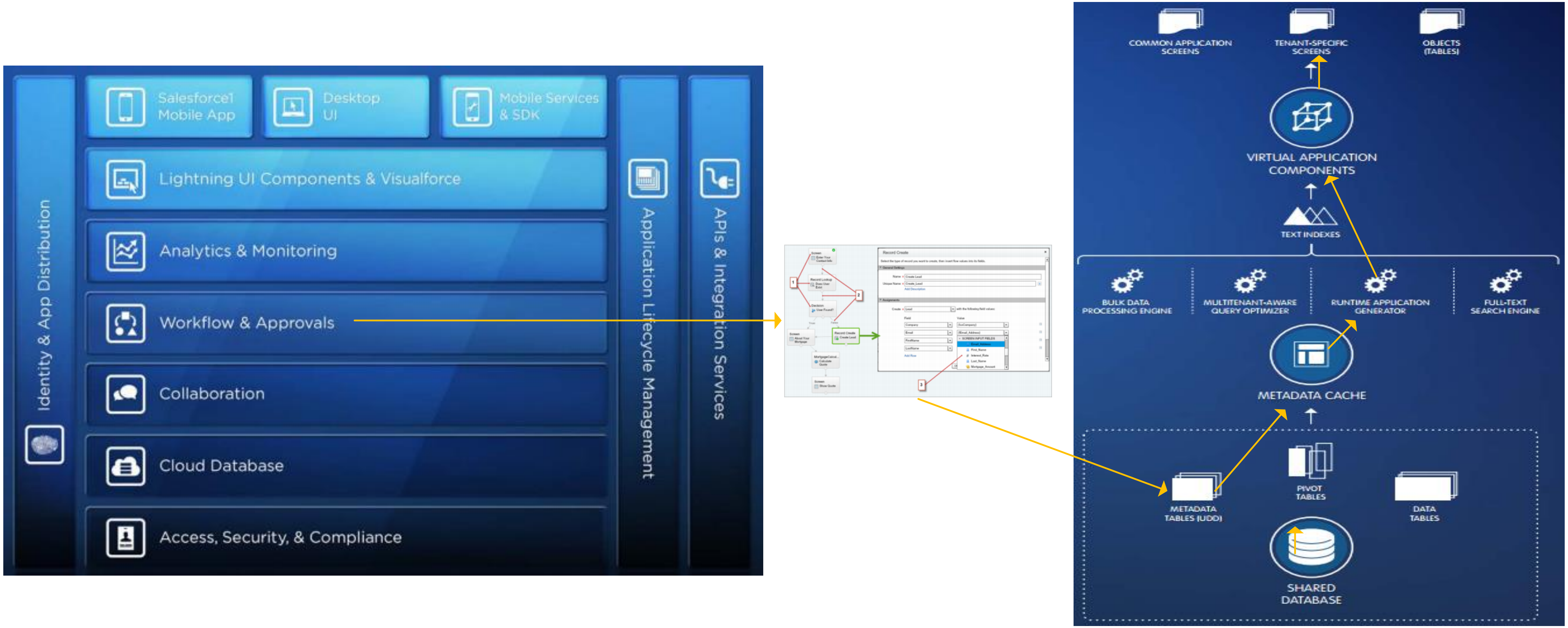
- Declaratively, using the native platform application framework
- Programmatically, using APIs, which are compatible with SOA and WOA development environments, including Visual Studio.NET (C#) and Apache Axis (Java and C++).



Salesforce.com, The Force.com Multitenant Architecture, 2008.

CASE STUDY: SAAS PLATFORM

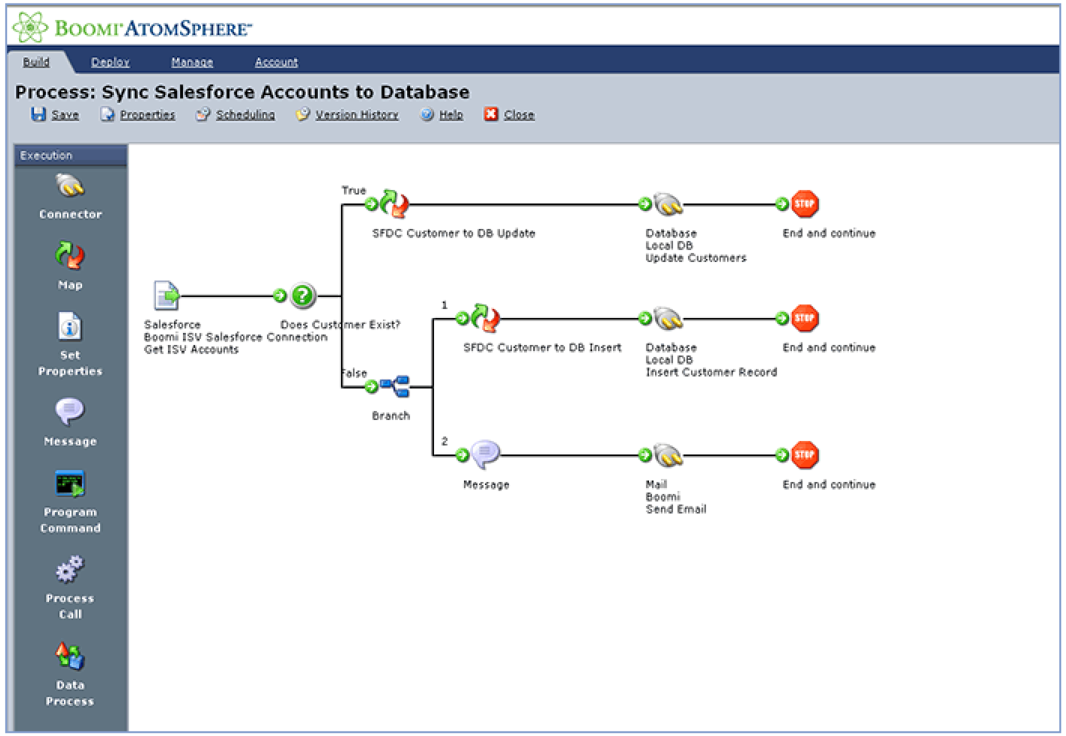
Salesforce 1 Platform


Salesforce1 Mobile App
Desktop UI
Mobile Services & SDK
Lightning UI Components & Visualforce
Analytics & Monitoring
Workflow & Approvals
Collaboration
Cloud Database
Access, Security, & Compliance
Identity & App Distribution
Application Lifecycle Management
APIs & Integration Services
Record Create
COMMON APPLICATION SCREENS
TENANT-SPECIFIC SCREENS
OBJECTS (TABLES)
VIRTUAL APPLICATION COMPONENTS
TEXT INDEXES
BULK DATA PROCESSING ENGINE
MULTITENANT-AWARE QUERY OPTIMIZER
RUNTIME APPLICATION GENERATOR
FULL-TEXT SEARCH ENGINE
METADATA CACHE
METADATA TABLES (UDD)
PIVOT TABLES
DATA TABLES
SHARED DATABASE

CASE STUDY: SAAS PLATFORM

Salesforce 1 Platform

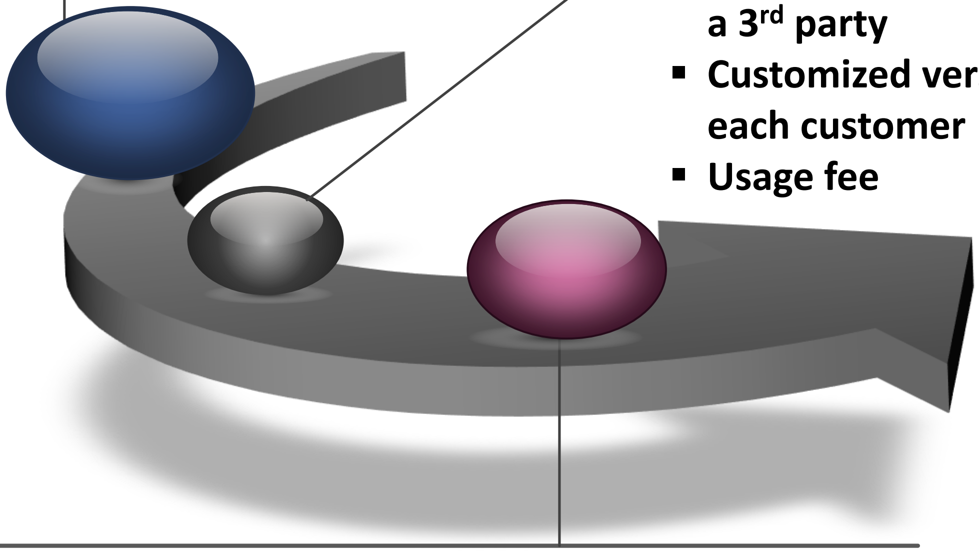
- On-demand integration solution handling SaaS-to-on-premise and SaaS-to-SaaS integration, targeting SaaS vendors and IT service vendors
  - Build – Design and configure your integration
  - Deploy – Run your integration anywhere
  - Manage – Centrally control your integrations
  - Embed – Productize integration to extend your application
  - Connector SDK – Develop & maintain your own connector
- Offers browser-based development and provisioning solution that uses Flash to run the Boomi IDE from a customer's browser



CASE STUDY: SAAS INTEGRATION PLATFORM

Boomi

EVOLUTION OF APPLICATION DELIVERY



Packaged Software

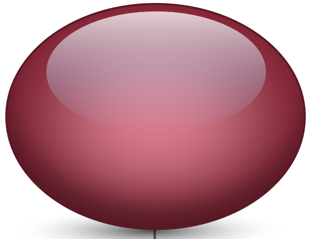
- Single code base with configuration parameters
- Single tenant
- Distribution of upgrades

ASP

- Packaged app hosted by a 3rd party
- Customized version for each customer
- Usage fee

Vendor-Managed Software (On-Demand Software Package)

- Single-tenant packaged app hosted by the app vendor
- License fee + maintenance/hosting Fee

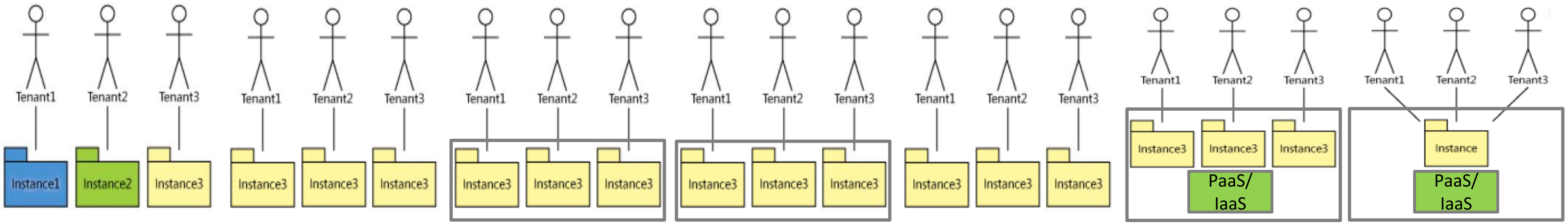


Software as a Service

- Single code base in multitenant architecture
- Frequent upgrade
- Automated ITSM over virtualization
- Usage fee
- Built-In monetization functions

Ted Ryan at OpSource says a multitenant application (as compared to a single-tenant) can reduce a SaaS provider's cost of goods sold (COGS) from 40 percent to less than 10 percent (since the cost of on-boarding a new customer approaches to zero, and upgrading is done on a single instance of the application).

EVOLUTION OF SOFTWARE PRODUCT



Licensing White-Box Package with Source Code changed for Customization (Multiple Code Bases)

Level 0

Licensing Configurable Application Package with Configuration Parameters for Customization (Single Code Base):

Level 1 (Packaged App)

Hosting Configurable Application Package by 3rd Party with CBOs for Customization (Single Code Base / Multiple App Instances)

Level 2 (ASP)

Licensing Composite Application Package in SOA (Single Code Base)

Level 3 (SOA Package)

Hosting Configurable Application Package Hosted by its Vendor (Single Code Base / Multiple App Instances)

Level 4 (Hosted App)

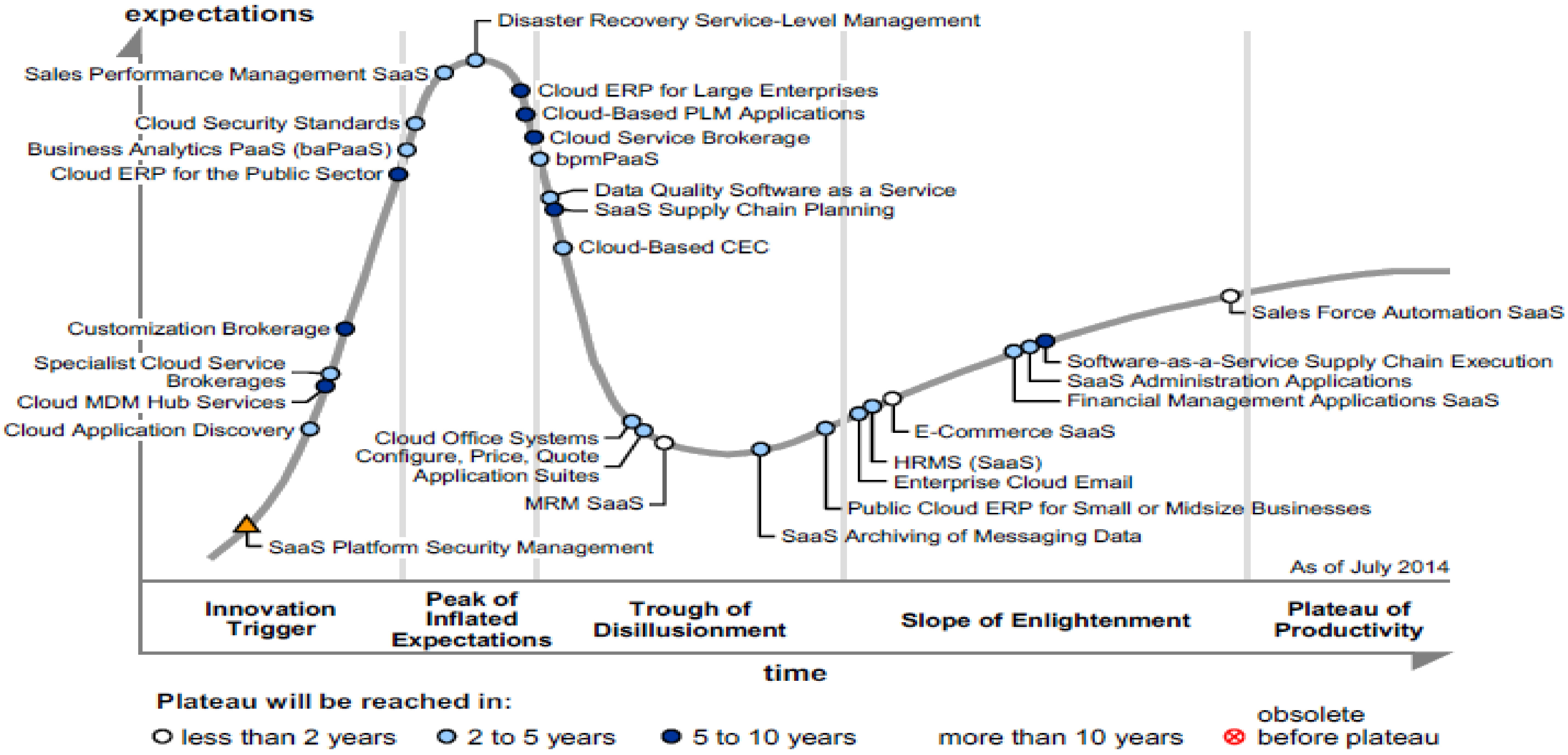
Single-Tenant SaaS on PaaS / IaaS (Single Code Base / Multiple App Instance)

Level 5 (Single-Tenant SaaS / PaaS)

Multitenant SaaS on PaaS / IaaS (Single Code Base / Single App Instance)

Level 6 (Multi-Tenant SaaS / PaaS)

SAAS HYPE CYCLE

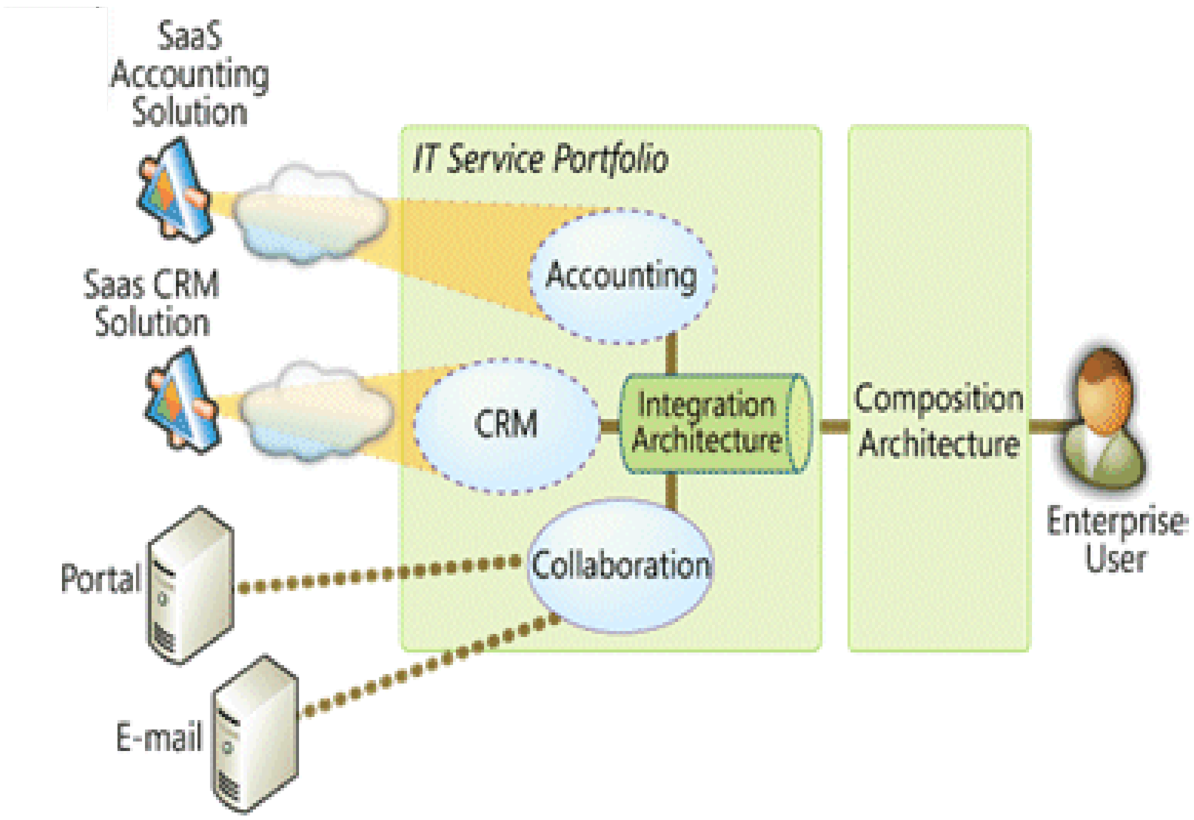


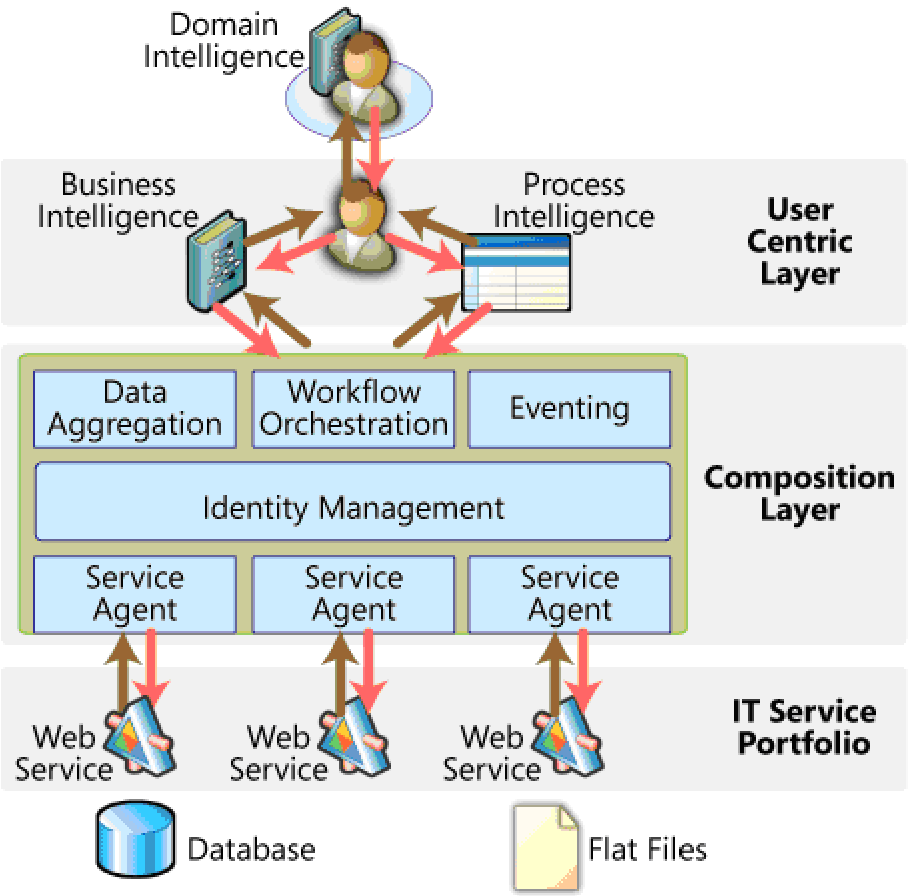
Gartner, Hype Cycle for Software as a Service, 2014

SAAS ARCHITECTURE

- Apply Service-Oriented Architecture (SOA) styles to SaaS backend design
  - Service-Oriented Architecture (SOA) is a natural and most used model for designing, implementing and consuming programmatic cloud services.
  - A cloud service appears as a SOA domain—an area of control where service name space, technology and use can be consistently governed—within the enterprise-wide application architecture.
  - Cloud computing and SOA both promote modularity, encapsulation, loose coupling, reuse and resource sharing. Growing adoption of cloud will further assert the core importance of SOA for mainstream business IT.
- Apply Web-Oriented Architecture (WOA) to SaaS frontend design
  - Provide "mashable" resources for enterprises including REST services, Web pages, social media, RSS/Atom feeds, POX, JSON, etc.
- Design SaaS-enabling attributes into the solution: multitenancy, scalability, tenant-specific security, availability and performance.
- Customizable GUI, data Model, Business Logic and Workflow.
- Be sure to design business 'Monetization' into the solution. For example Gold, Silver and Bronze levels of service based upon functionality, usage, entities, transactions, consumption etc.

SERVICE-ORIENTED, ENTERPRISE SAAS ARCHITECTURE





Microsoft, Software as a Service: Enterprise Perspective, 2006.