

CA Layer7 - API Management

Product Features, Strategies and Analytics

Sreevidya Poola

APIM - Developer

Miracle Software Systems, Inc.

Agenda

- The **Miracle** Story
- What is **API**?
- **CA API** Management
- **Product** Family
- **Components** and **Types of Gateways**
- **Architecture, Strategies** and **Analytics**
- **Deployment** forms
- **Questions** and **Answers**

The **Miracle** Story



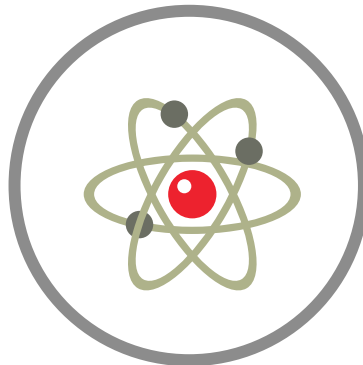
**Customer
Satisfaction**



**Affordable
Innovation**



**Quality and
Efficiency**



**Talent
Eco-System**

12
Global
Locations

15+
Technology
Partnerships

25+
Technology
Accelerators

20+
Years of
Expertise

500+
Certified
Professionals

85+
Fortune 1000
Customers

2000+
Employees
Globally

Miracle's Core Industry Expertise



Large pool of certified resources in SAP, IBM, Microsoft and more

Certified Resources



Numerous accelerators aimed at bringing agile adoption to technology

Reusable Frameworks



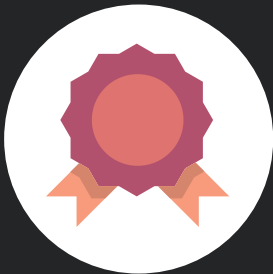
Application Delivery/Testing(Offshore/Offsite/Nearshore)

Global Delivery



Resources provided at no cost to mitigate resource turnover

Shadow Resources



For Bugs and Fixes after completion of the development

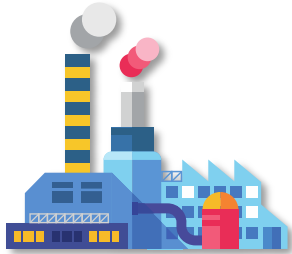
60 Day Warranty



Post-Prod maintenance and support option at fixed cost per ticket

Production Support

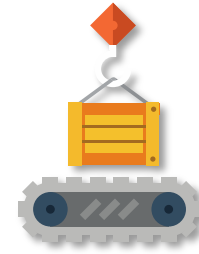
Industry Verticals



**Manufacturing
and Automotive**



**Energy and
Utilities**



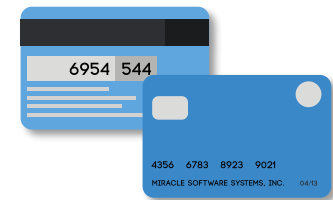
**Logistics and
Supply Chain**



**Healthcare and
Pharmacy**



**Retail
Industry**



**Banking and
Insurance**

Strategic Partnerships



Over the years we have formed numerous
strategic partnerships to ensure that we can
serve our customers right



What is **API**?

- Application Programming Interface (API), is a set of protocols, and tools for building **Software Applications**
- APIs express a **Software Component** in terms of its operations, inputs, and outputs
- APIs often come in the form of a library
- Application developers calls these API functions and use them by defined protocols like **Soap**, and **Rest Architecture** style

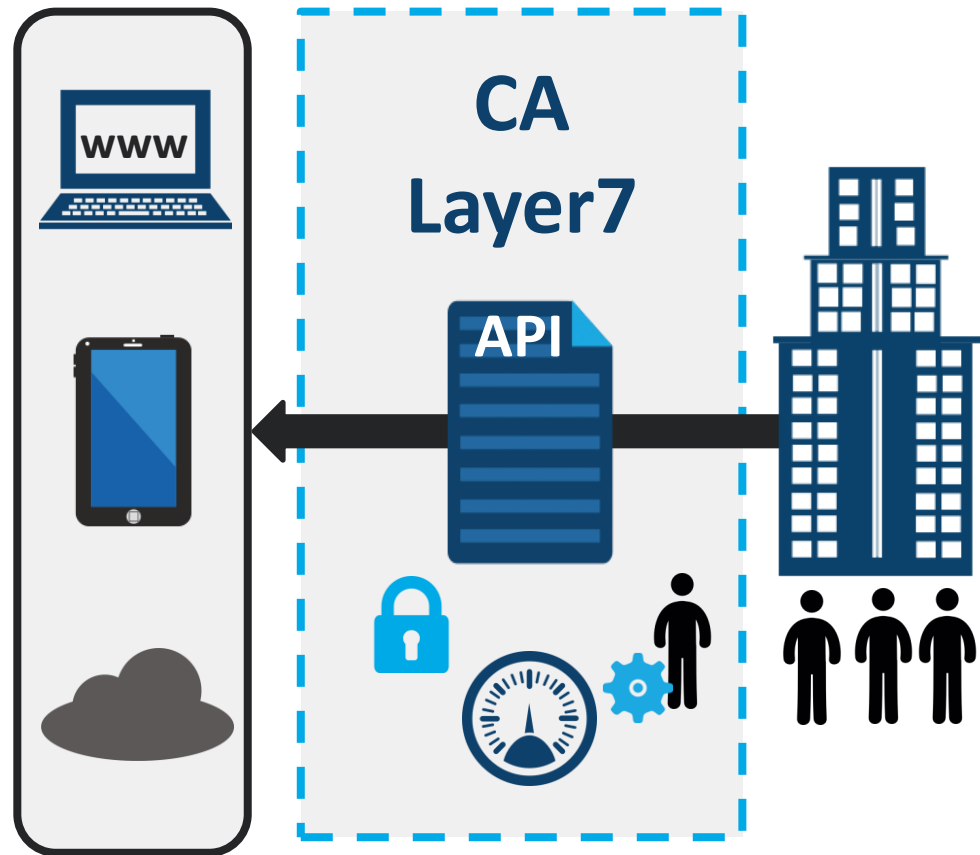


CA API Management

- APIs provide the connectivity to meet these demands and are the building blocks of today's enterprise growth strategies
- With CA API Management, you'll acquire the capabilities you needed to overcome digital transformation challenges and create new growth opportunities. The following are some of them,
 - Create APIs and Integrate Everything
 - Secure the Open Enterprise
 - Accelerate Mobile and IoT Development
 - Unlock the Value of Data

CA API Management product family

- API Gateway
- Mobile API Gateway
- Live API Creator
- API Developer portal
- API Management SaaS
- Mobile APP Services





Product Components

API PORTAL

1

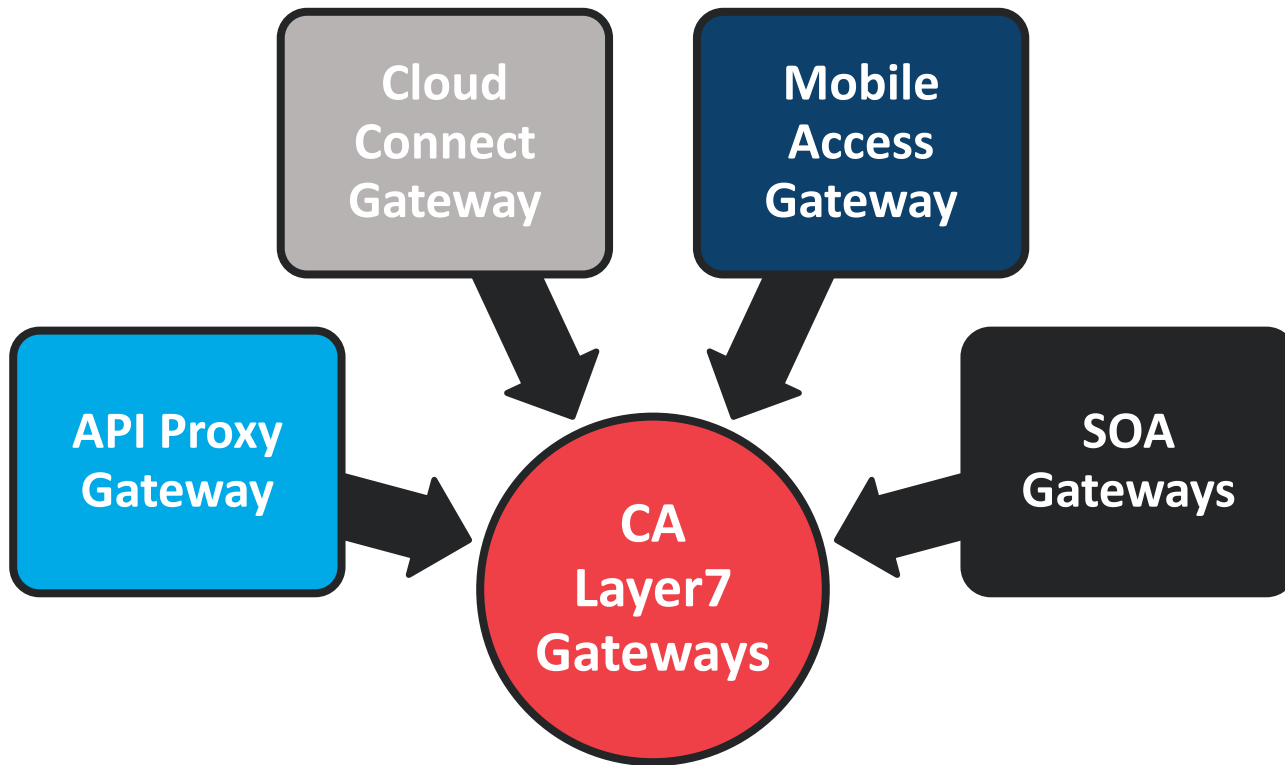
GATEWAY

2

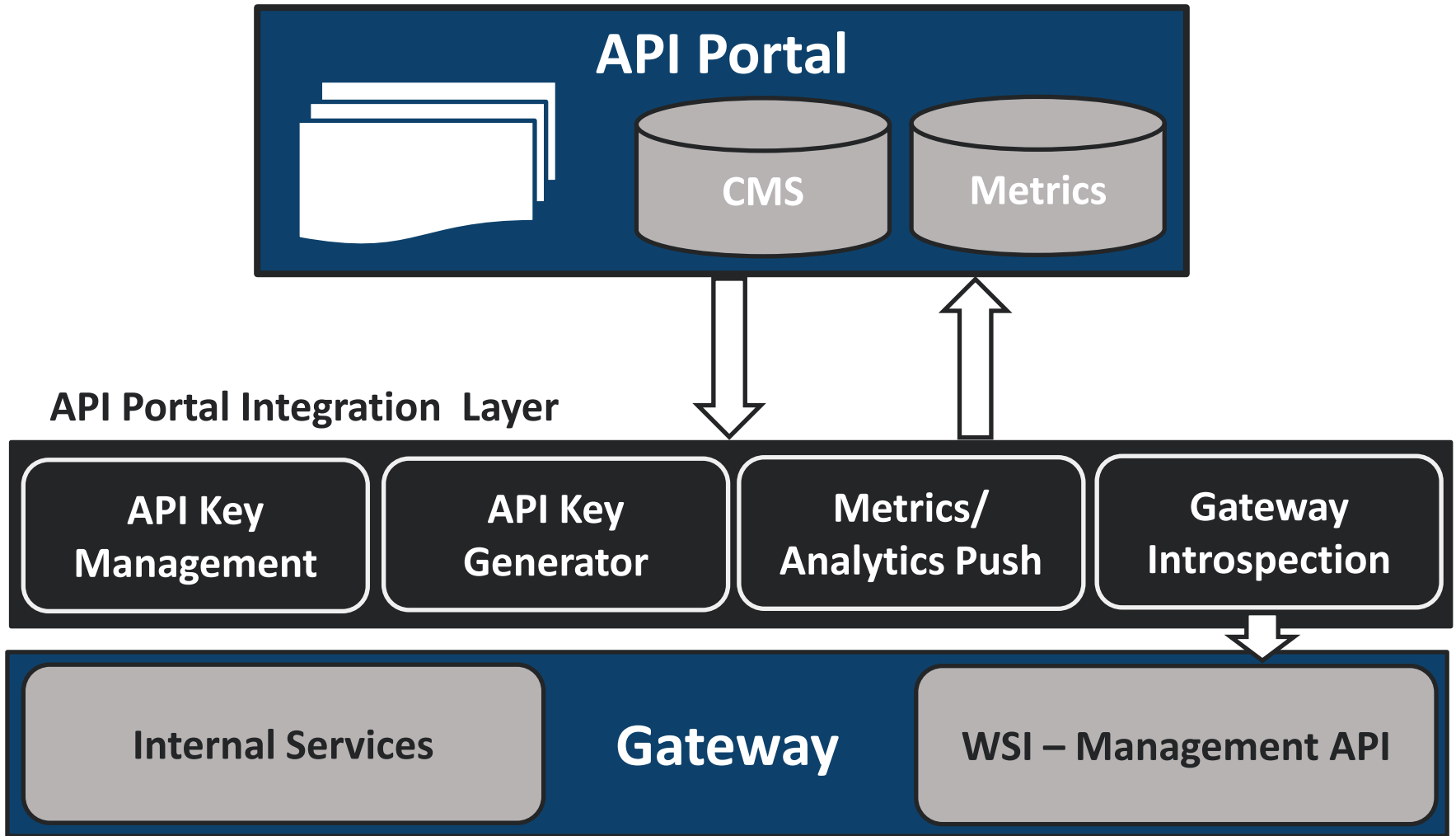
ESM

3

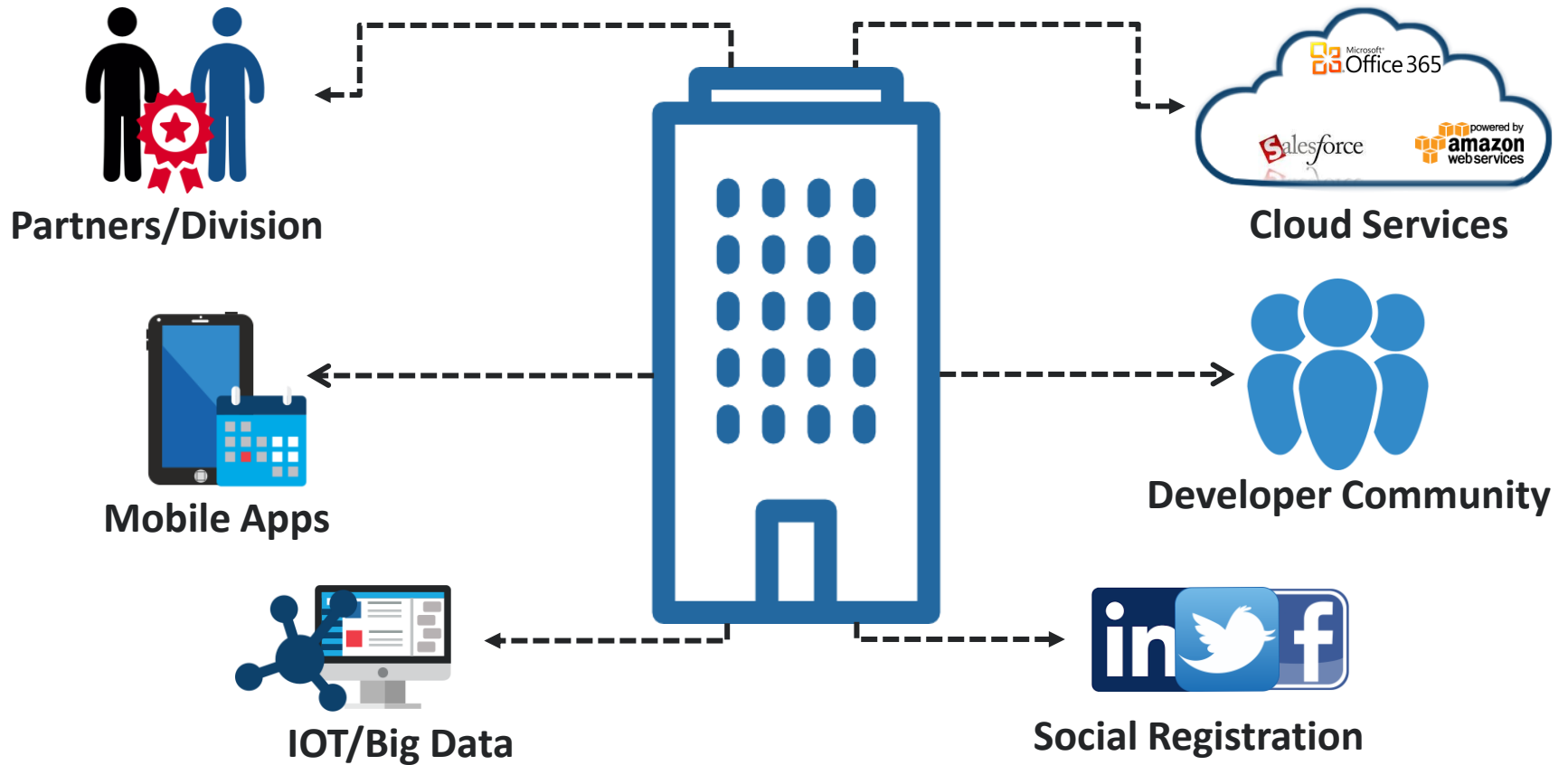
Types Of Gateways



Architecture



Open Enterprise in CA API Management



Analytics for Open Enterprise



Partners/Division



Mobile Apps



IOT/Big Data

79%

Of Organizations are
using SaaS3

1.43B

Of Organizations are
using SaaS3

305B

Mobile App
downloads by 2016

50B

Connected Devices
by 2020

35ZB

Of data BY 2025



Cloud Services



Developer Community



Social Registration

CA Software - Manage and Secure APIs

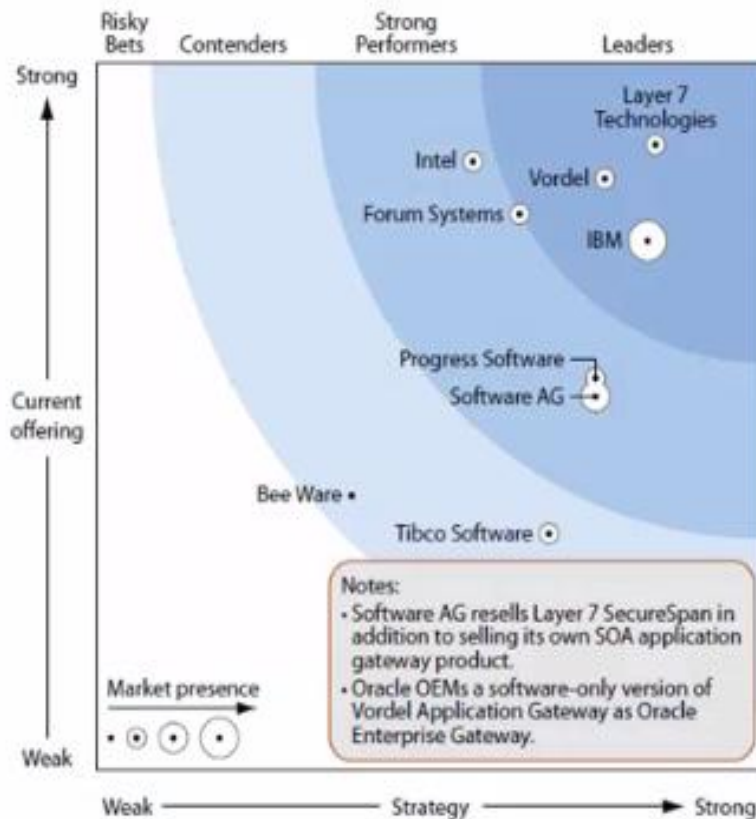


Backend Data into APIs



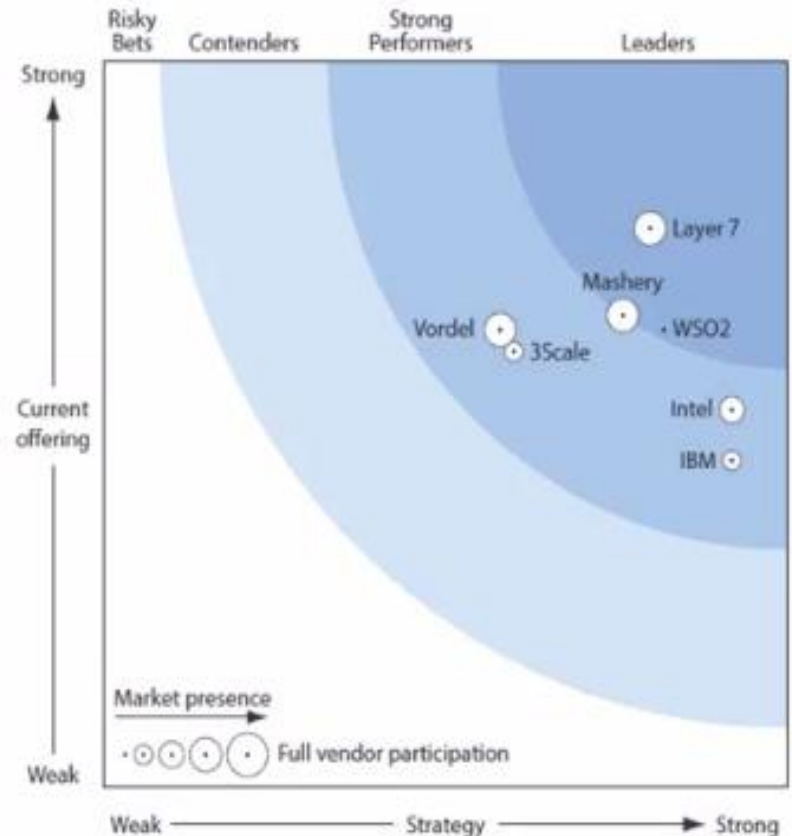
Why is CA at the Top of Industry?

The Forrester Wave:
SOA Application Gateways, Q4 2011



Forrester Research Inc., "The Forrester Wave: SOA Application Gateways", Q4 2011, November 18, 2011

Forrester Wave:
API Management Platforms, Q1 2013



Forrester Research Inc., "Forrester Wave: API Management Platforms, Q1 2013", February 5, 2013

Market Share for CA



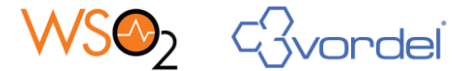
Differences Between Platforms



- Incredible track record of POC Success
- Easy to install and demo
- Deeper than the competition



- IBM's solution requires 3 or more products
- The layer7 team has deep experience on how to work with and defeat IBM Middleware



- Layer7 has the most complete product offering and the best enterprise pedigree to beat other API Management vendors in Top Accounts

Layer7 API Main Core Competitors Apigee

Why do They Sell

- Cloud based API Management solution, capacity based pricing
- On-premise (custom) solution

Whom Does They Sell

- App Developers
- Especially targeting telcos, carriers
- Doesn't target operations and IT

What are They Known For

- Good looking GUI, well packed
- Also have a free version
- Very limited Integration and Security functionality



Layer7 SOA Core Competitors

Why do They Sell

- A physical set of devices to handle SOA integration and security
- They recently introduced a virtual appliance

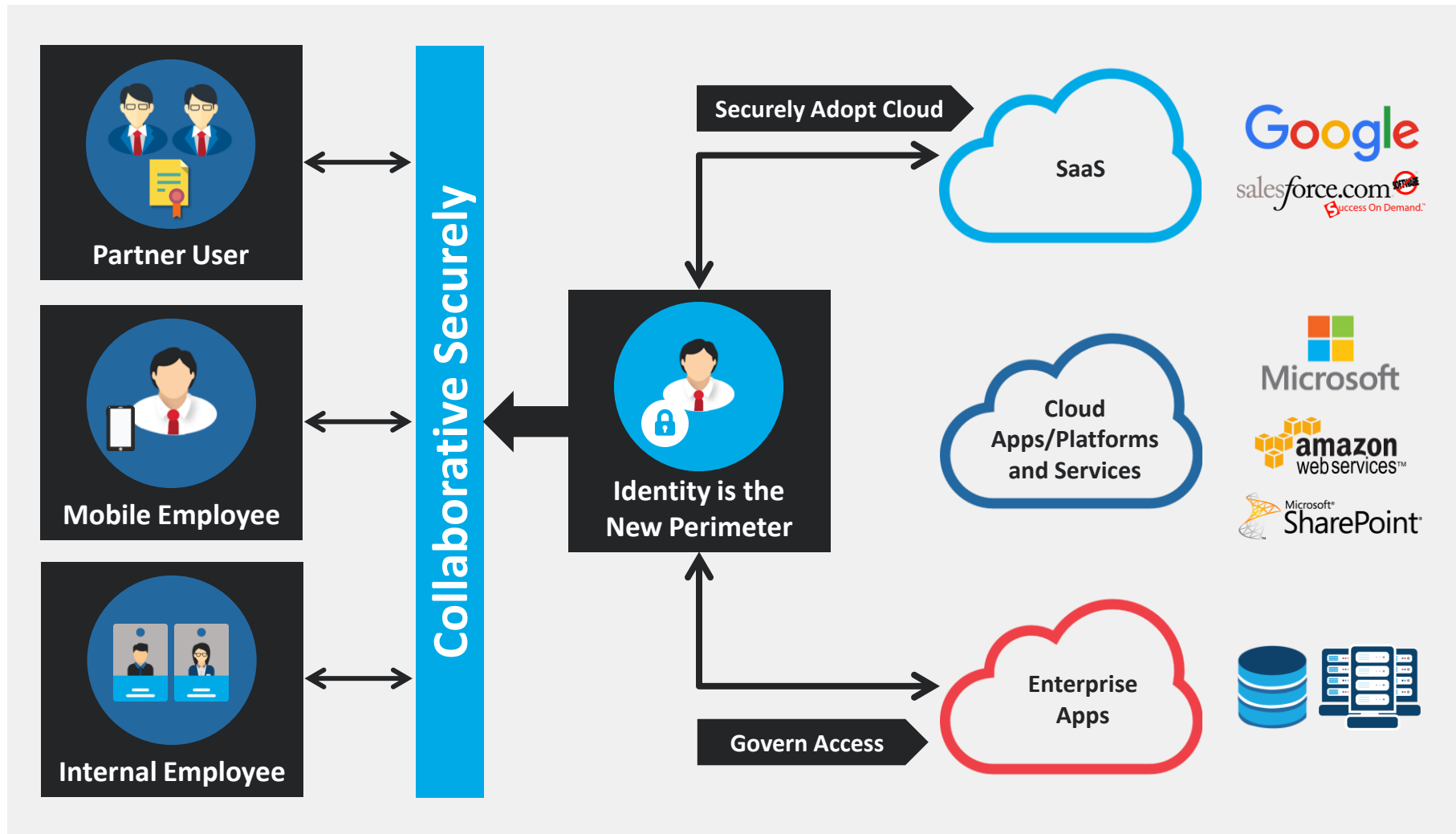
Whom Does They Sell

- Enterprise and Infrastructure Architects for SOA
- Not sold as much to Security teams(In WebShpere, not Tivoli)

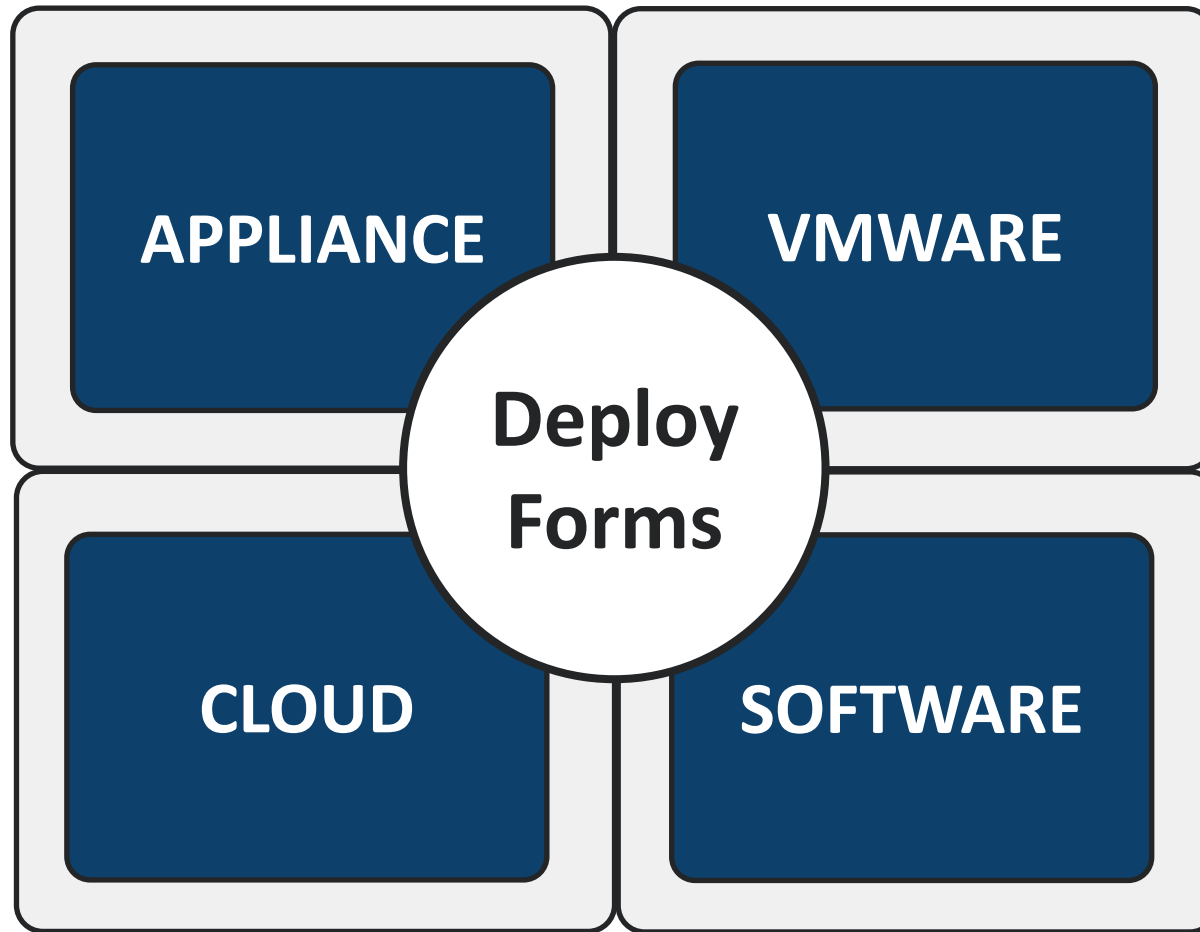
What are They Known For

- XML processing, Web Services, acceleration
- Not known for flexibility and product features have changed little recently

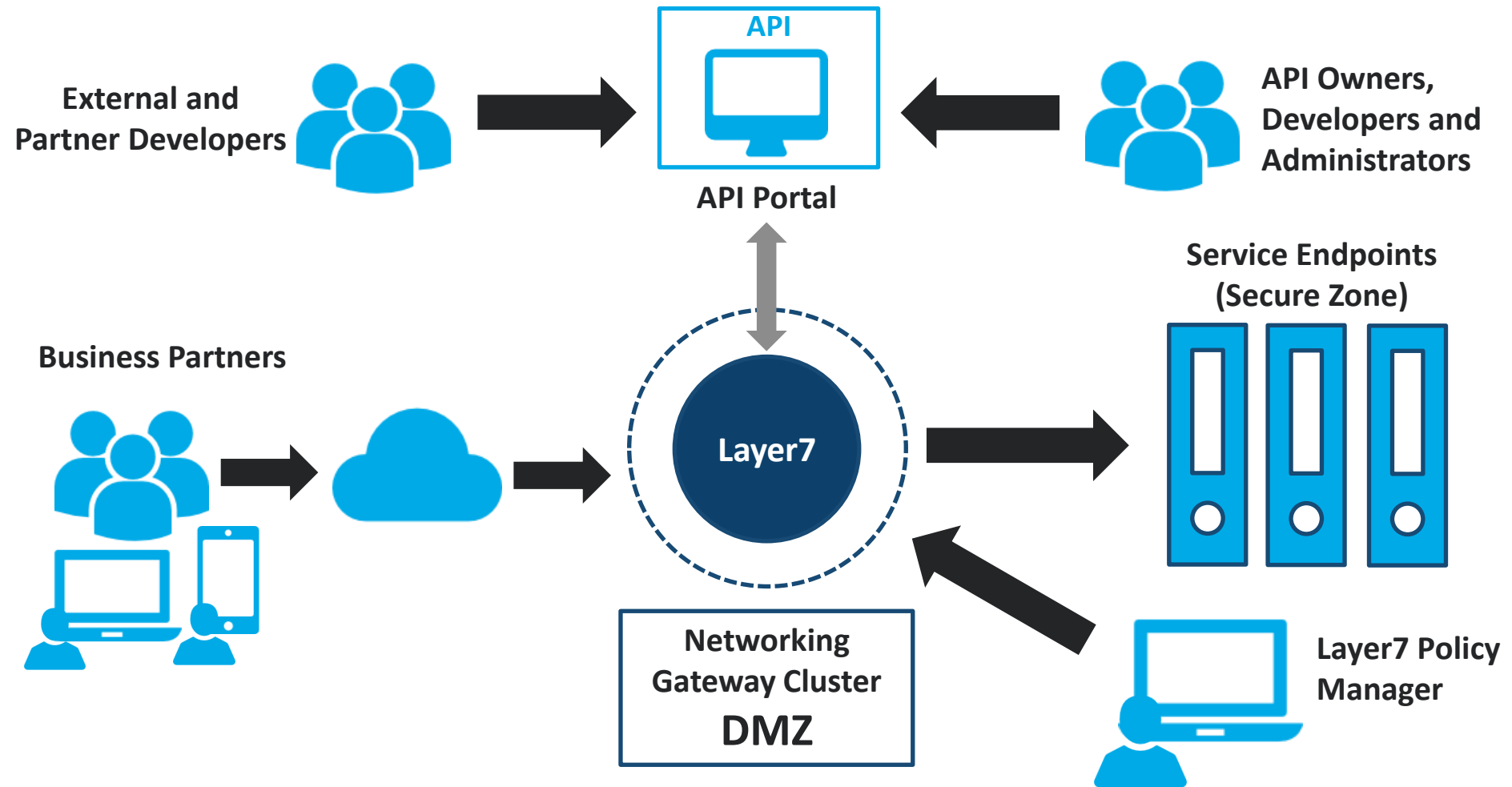
Secure Mobile and Cloud Connected Enterprise



Various Deploy Forms



API Portal and Gateway





Features of the Gateway

1 Security

2 Performance & Scale

3 Manageability

4 Flexibility & Integration

5 Extensibility

API Portal



Developer Network

[Login](#) | [Signup](#)



[Home](#)

[Knowledge Base](#)

[Resources](#)

[Forums](#)

[Home](#) > [Knowledge Base](#) > [Troubleshooting](#)

KNOWLEDGE BASE

SETUP / CONFIGURATION

SITE CUSTOMIZATION

TROUBLESHOOTING

FAQ

Portal-Gateway Communication Troubleshooting

Passwords

When troubleshooting communication between a portal and gateway instance, it is necessary to ensure that all passwords being use are correct. This can be accomplished with the following:

1. Gateway Password

- Login to the Policy Manger using the /apiportal account (ensure the account has administrator privileges)

2. Keystore Password

- Issue the following openssl command on the exported private key .p12 file:

```
openssl pkcs12 -info -in <your_exported_private_key>.p12
```

3. Truststore Password

- From within the /opt/Deployments/lrs/server/conf/keys/ directory:

```
/opt/jdk/bin/keytool -list -keystore trustedCerts.ks -storepass <truststore_password>
```

If any of the 3 passwords above (i.e., Gateway, Keystore, Truststore) are incorrect, the correct passwords will need to be re-entered into the Layer 7 Gateway Plugin in the CMS.

**important note about the truststore password:

If the truststore password has been changed from the default of "changeit" the change must be reversed. There is a bug related to this password requiring that it not be modified. If it has been changed, the following command to reset it back to "changeit" from within the /opt/Deployments/lrs/server/conf/keys/ directory:

Policy Manager

The screenshot displays the Policy Manager interface. On the left, the 'Assertions' tab is active, showing a list of assertion types. The 'Identity Providers' tab is also visible. The main pane shows the 'API Key Management Service [/api/keys/*] (v1/1, active)' policy. The policy is composed of several assertions, many of which are marked with green checkmarks, indicating they are active or successful. The assertions include:

- At least one assertion must evaluate to true
 - All assertions must evaluate to true
 - Require SSL or TLS Transport with Client Certificate Authentication
 - Add Audit Details: "username: \${request.username}"
 - Request: Authenticate against Internal Identity Provider
- All assertions must evaluate to true
- Audit Messages in Policy
 - Add Audit Details: "auth_user: \${request.authenticateduser}"
 - Set Context Variable uri.generate as String to: /generate
 - Set Context Variable uri.register as String to: /register
 - Set Context Variable uri.update as String to: /update
 - Set Context Variable uri.delete as String to: /delete
 - Set Context Variable errorCode as String to: 0
 - Comment: // Setup debugging portion (only when enabled)
- At least one assertion must evaluate to true
 - All assertions must evaluate to true
 - Compare Variable: \${request.http.parameter.debug} is a String and is equal to true (case sensitive); If Multivalued
 - Set Context Variable debugBlock as String to: <!-- debug block --> <l7:Debug xmlns:l7="http://ns.l7te...
 - Set Context Variable debugBlock as String to empty
- At least one assertion must evaluate to true
 - All assertions must evaluate to true
 - Compare Variable: \${request.http.uri} is a String and contains \${uri.generate} (case sensitive); If Multivalued
 - Set Context Variable serviceOp as String to: \${request.http.uri}
 - Set Context Variable validServices as Message to: \${gateway.portal.managed.services}
- At least one assertion must evaluate to true
 - All assertions must evaluate to true
 - Compare Variable: \${errorCode} is a String and is not equal to 0 (case sensitive); If Multivalued all v
 - Set Context Variable fragmentResponse as Message to: <l7:ApiKeyResponse xmlns:l7="http://ns.l7te...
 - Continue Processing
- All assertions must evaluate to true
- At least one assertion must evaluate to true
 - All assertions must evaluate to true
 - Compare Variable: \${errorCode} is a String and is equal to 0 (case sensitive); If Multivalued all values must
- Return Template Response to Requestor
- All assertions must evaluate to true

Policy Manager viewing context variables

The screenshot displays the Policy Manager interface for the 'API Key Management Service [/api/keys/*] (v1/1, active)'. The left pane shows a tree of assertions, including 'At least one assertion must evaluate to true', 'All assertions must evaluate to true', 'Require SSL or TLS Transport with Client Certificate Authentication', 'Add Audit Details: "username: \${request...}', 'Request: Authenticate against Interna...', 'Set Context Variable uri.generate as String to...', 'Set Context Variable uri.register as String to...', 'Set Context Variable uri.update as String to...', 'Set Context Variable uri.delete as String to...', 'Set Context Variable errorCode as String to...', 'Comment: // Setup debugging portion (only...', 'At least one assertion must evaluate to true', 'All assertions must evaluate to true', 'Compare Variable: \${request.http.pa...', 'Set Context Variable debugBlock as S...', 'Set Context Variable debugBlock as Strin...', 'At least one assertion must evaluate to true', 'All assertions must evaluate to true', 'Compare Variable: \${request.http.uri...', 'Set Context Variable serviceOp as St...', '\${serviceOp}: Evaluate Regular Expr...', 'Set Context Variable validServices as...', 'At least one assertion must evaluate...', 'At least one assertion must evaluate...', 'All assertions must evaluate to tr...', 'Compare Variable: \${errorCod...', 'Set Context Variable fragment...', 'Continue Processing', 'All assertions must evaluate to true', 'At least one assertion must evaluate to true', 'All assertions must evaluate to true', 'Compare Variable: \${errorCode} is a...', 'Return Template Response to Rec...', and 'All assertions must evaluate to true'.

The right pane shows the 'Context Variable Properties' dialog box for the variable 'debugBlock'. The 'Variable Name' is 'debugBlock', the 'Data Type' is 'String', and the 'Content-Type' is 'text/xml; charset=utf-8'. The 'Expression' is:

```
<!-- debug block -->
<l7:Debug xmlns:l7="http://ns.l7tech.com/2011/08/portal-api-keys">
  <l7:Method>${request.http.method}</l7:Method>
  <l7:Uri>${request.http.uri}</l7:Uri>
  <l7:HttpHeaders>request.http.allheadervalues</l7:HttpHeaders>
  <l7:HttpParams>
    <l7:Param name="apikey">${request.http.parameter.apikey}</l7:Param>
    <l7:Param name="apikeySecret">${request.http.parameter.apikeySecret}</l7:Param>
    <l7:Param name="apikeyStatus">${request.http.parameter.apikeyStatus}</l7:Param>
    <l7:Param name="assignApis">${request.http.parameter.assignApis}</l7:Param>
    <l7:Param name="format">${request.http.parameter.format}</l7:Param>
    <l7:Param name="debug">${request.http.parameter.debug}</l7:Param>
  </l7:HttpParams>
</l7:Debug>
```

The 'Save line breaks as' options are 'CR LF' (selected), 'LF', and 'CR'. The 'OK' button is highlighted.

API Management Enables Digital Access



Integrate Everything

- Expose legacy system and applications as APIs, allowing them to be consumed in new ways
- Integrate your digital projects into one project for easier management

CA API Gateway
CA Mobile API
Gateway



Enable Developers

- Provide developers with APIs to build apps that deliver a great user experience
- Manage developers through Marketing, Onboarding, Collaboration and Testing
- Manage access to sensitive enterprise data

CA API Developer
Portal



Protect Apps

- Guard against inherent risk of exposing APIs
- Protect your apps with end-to-end military grade service
- Control which specific data your partners and developers can access

CA OAuth Toolkit
CA Mobile API
Gateway



Monetize API's

- Plan digital models in a flexible way and generate revenue for the use of APIs
- Analyze and report on API activity

CA API Developer
Portal

Questions and Answers Time!!

We're more than happy to help out

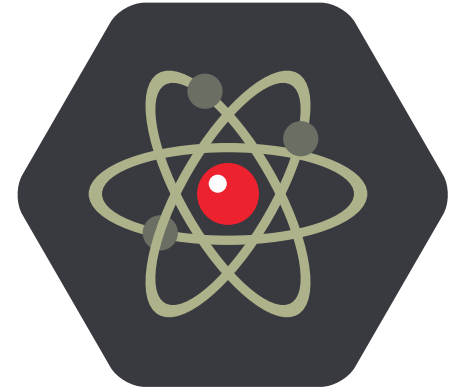
Digital Transformation and Miracle



Health Check
Workshops



Digital Transformation
Workshops



Labs and COE
Pilot Services

Our **Innovation Labs** and **COE** Teams can help you with a pilot (or) help build a Road Map for your nextGen **Digital Transformation** initiatives.

Thank You

Our teams are dedicated to innovating with IT and redefining solutions for customer excellence.

To learn more visit,
www.miraclesoft.com



@Team_MSS



/miracle45625



/miracle45625