



Testing anything, any time
with containerized service virtualization

Leveraging on-demand virtual test environments
for earlier, better and faster testing

Robert Schrijvers, Ruben Wildeboer & Rix Groenboom

Who are we

oint work:

Rix Groenboom (Parasoft EMEA)

Ruben Wildeboer (Parasoft NL)

Robert Schrijvers

- Independent Consult
- Java specialist with focus on performance
- Working together with Parasoft since 2012
- Working for various banks in the Netherlands

Company Highlights



Where/What/How can Parasoft help?

Agile + DevOps

Safety Critical

Continuous Testing

Development Testing

Development Testing

Test Environment Isolation

Test collaboration

API/Message Layer Testing

Accelerate Agile

Compliance

Embedded Testing

Agenda

Modern times and need for Virtualization

Virtualization per phase

- Development
- Test
- Acceptance
- Pre-production

Requirements & implementation

- Functionality (Dev)
- Deployment (Ops)

Wrap up

Agenda

Modern times and need for Virtualization

Virtualization per phase

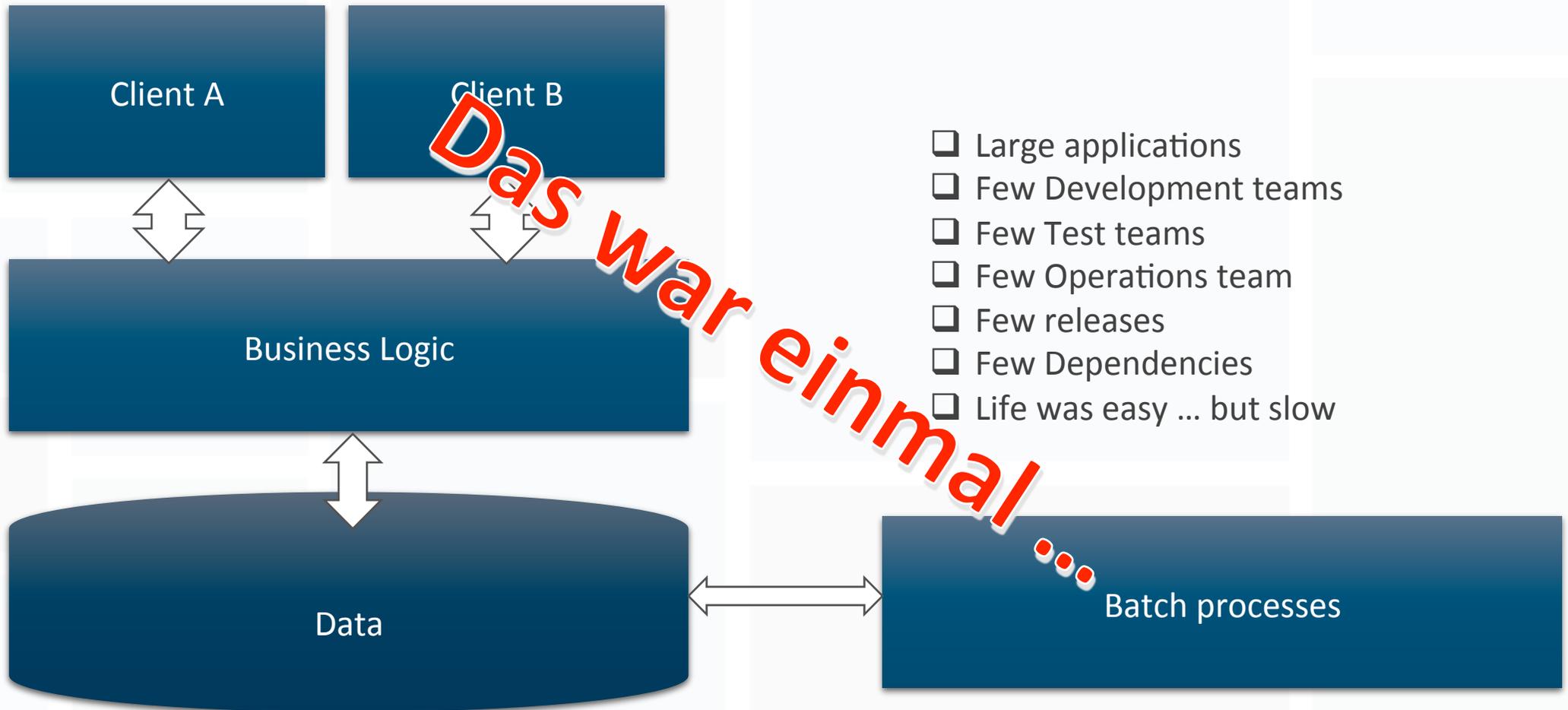
- Development
- Test
- Acceptance
- Pre-production

Requirements & implementation

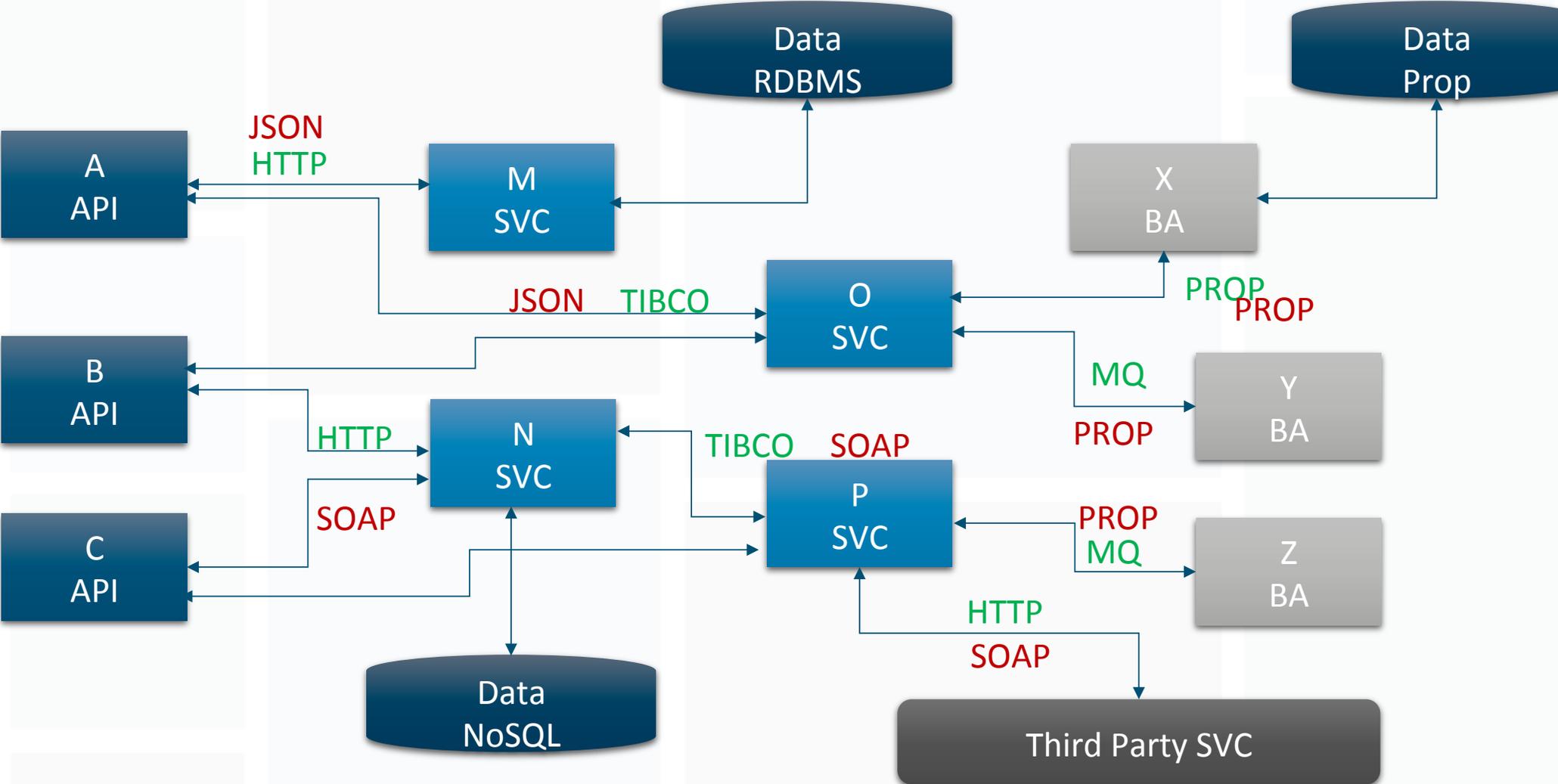
- Functionality (Dev)
- Deployment (Ops)

Wrap up

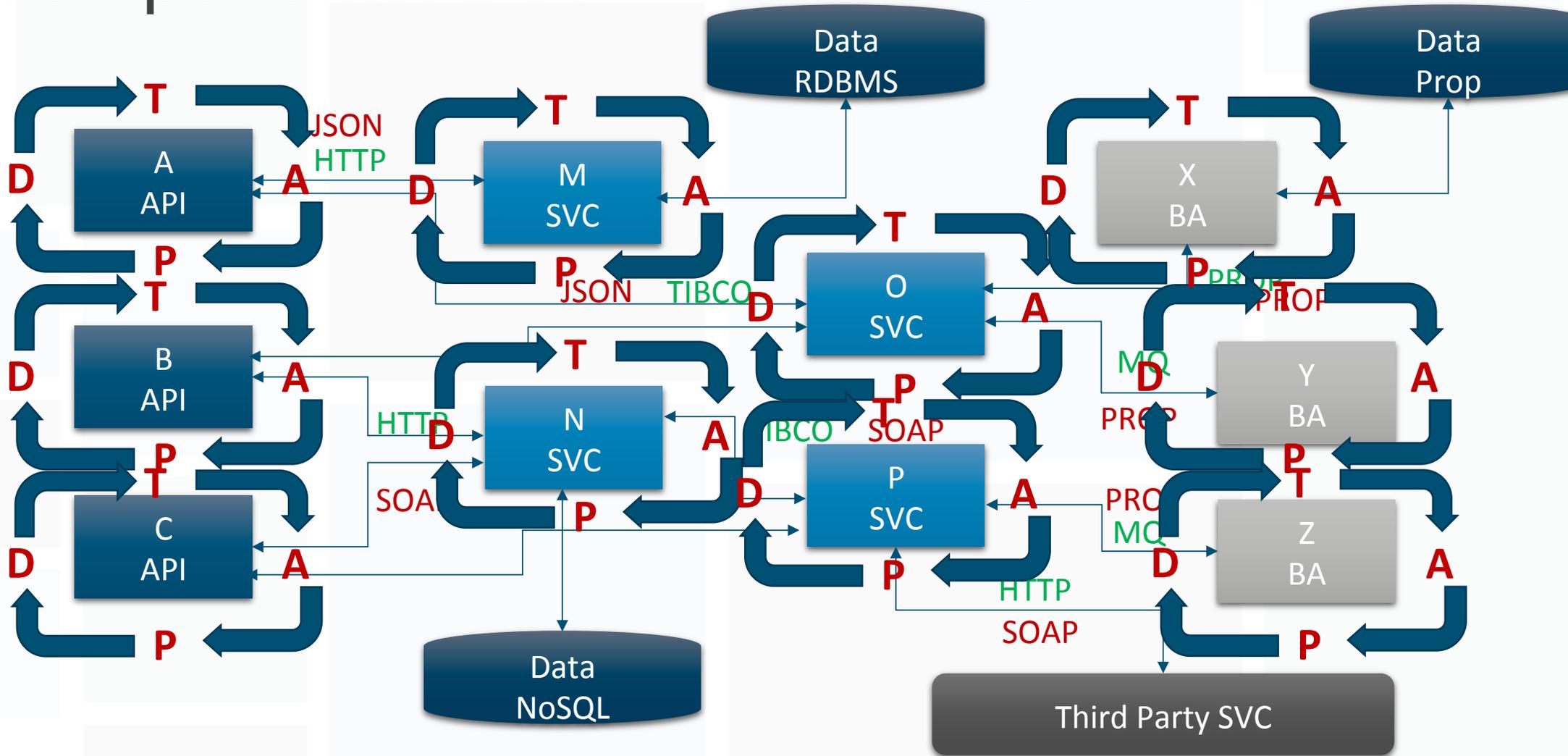
Once upon a time ...



Modern times



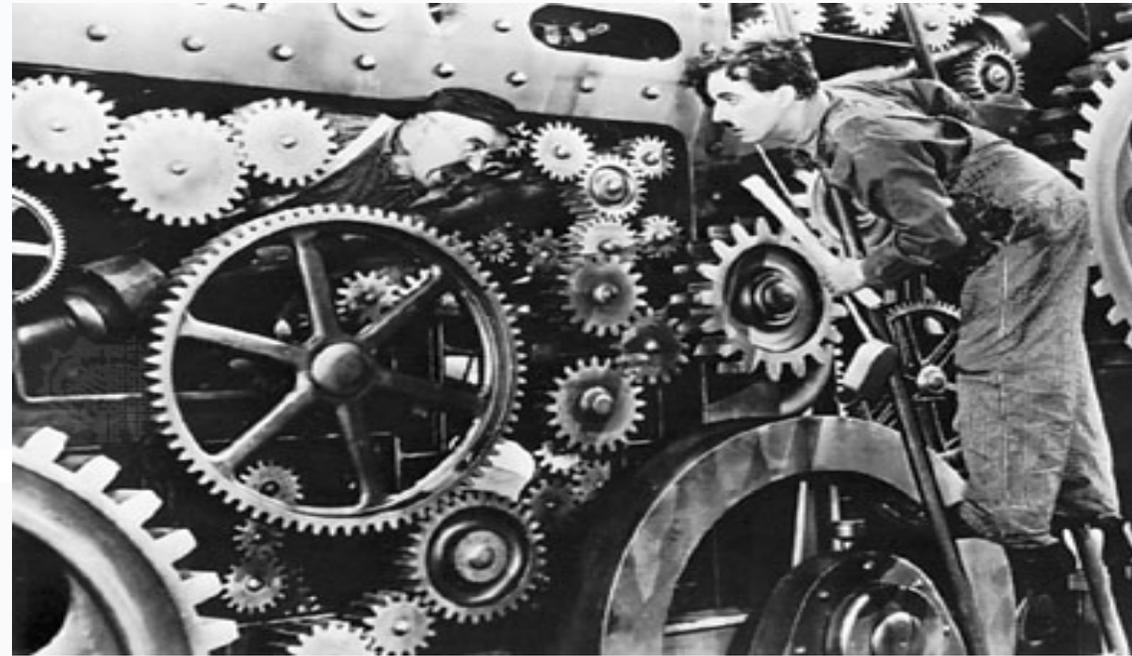
Components: Dev -> Tst -> Acc -> Prod



DevOps teams

- Many teams
 - Independent
 - Different cycle speed
-
- Test often
 - Test early
 - Test automatically

“Modern times”



Decoupling by virtualization

- Prevent teams slowing down by other teams
- Prevent skipping tests in early stages
- Extend test cases

- Depends on stage in release cycle
- Various performance profiles
- Various deployment types
- Various transports (protocols)
- Various payloads (messages-types)

Agenda

Modern times and need for Virtualization

Virtualization per phase

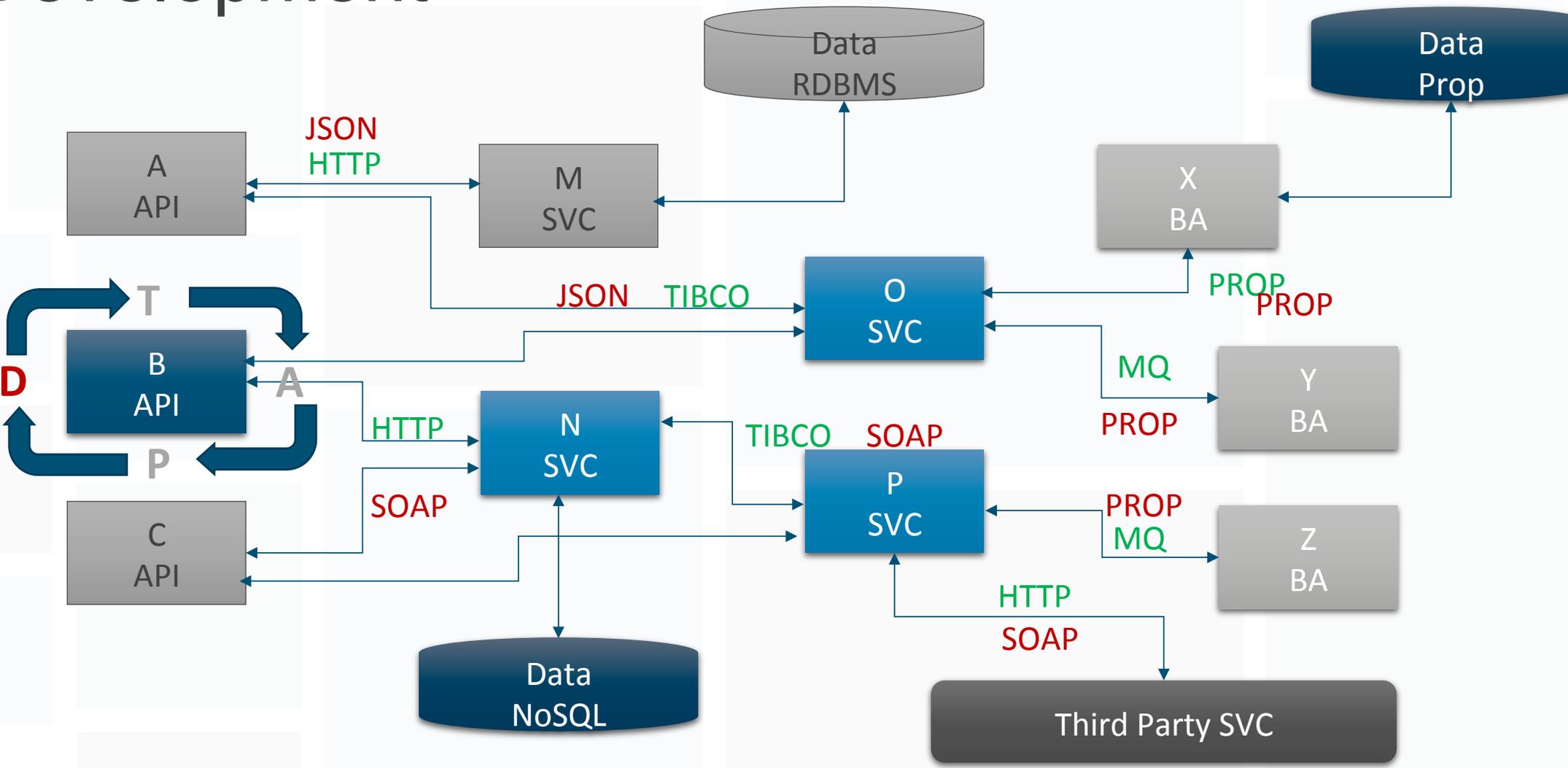
- Development
- Test
- Acceptance
- Pre-production

Requirements & implementation

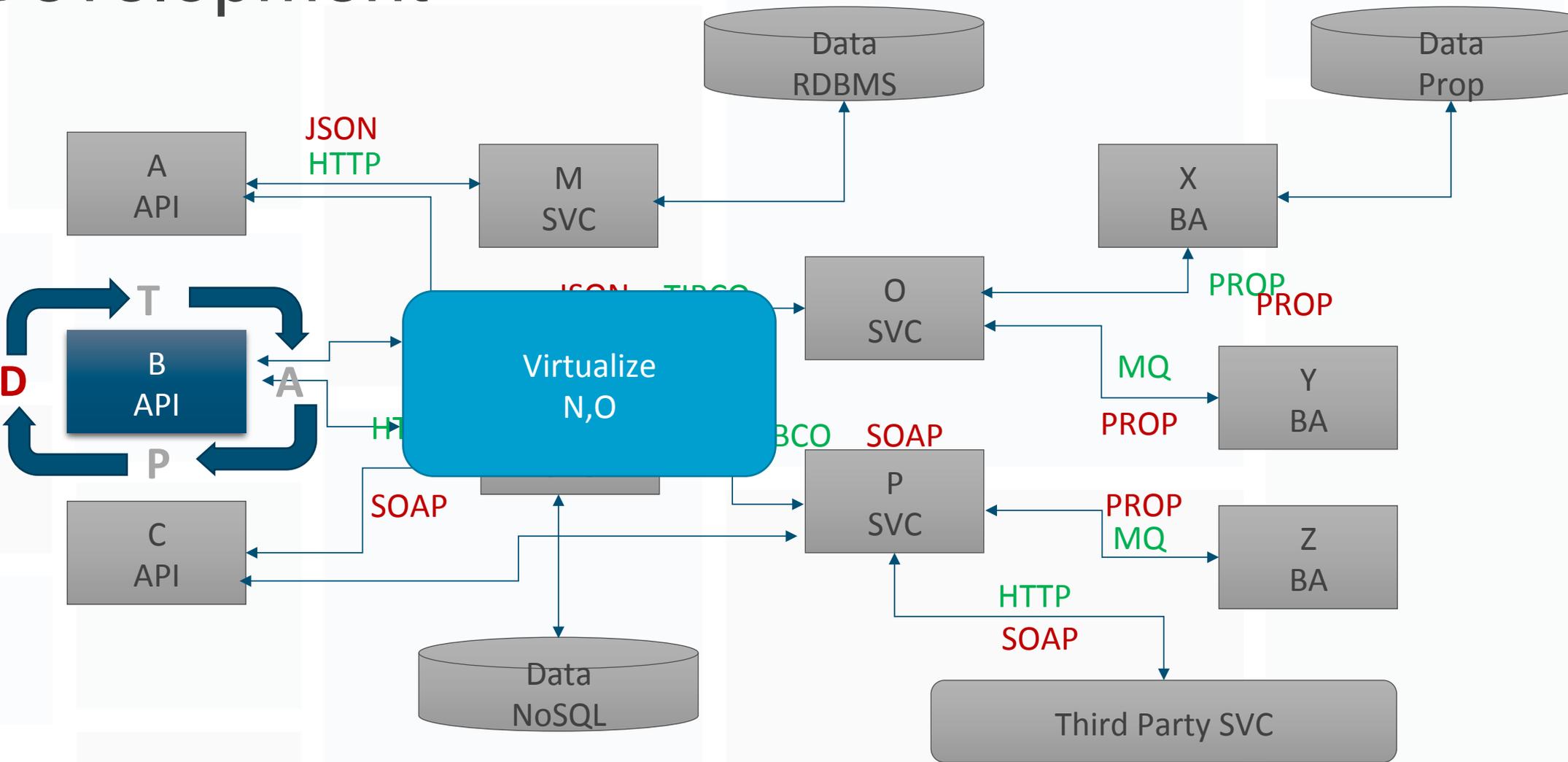
- Functionality (Dev)
- Deployment (Ops)

Wrap up

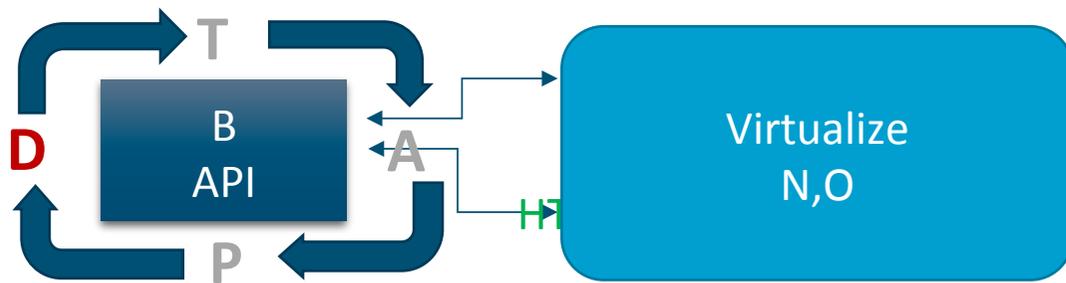
Development



Development

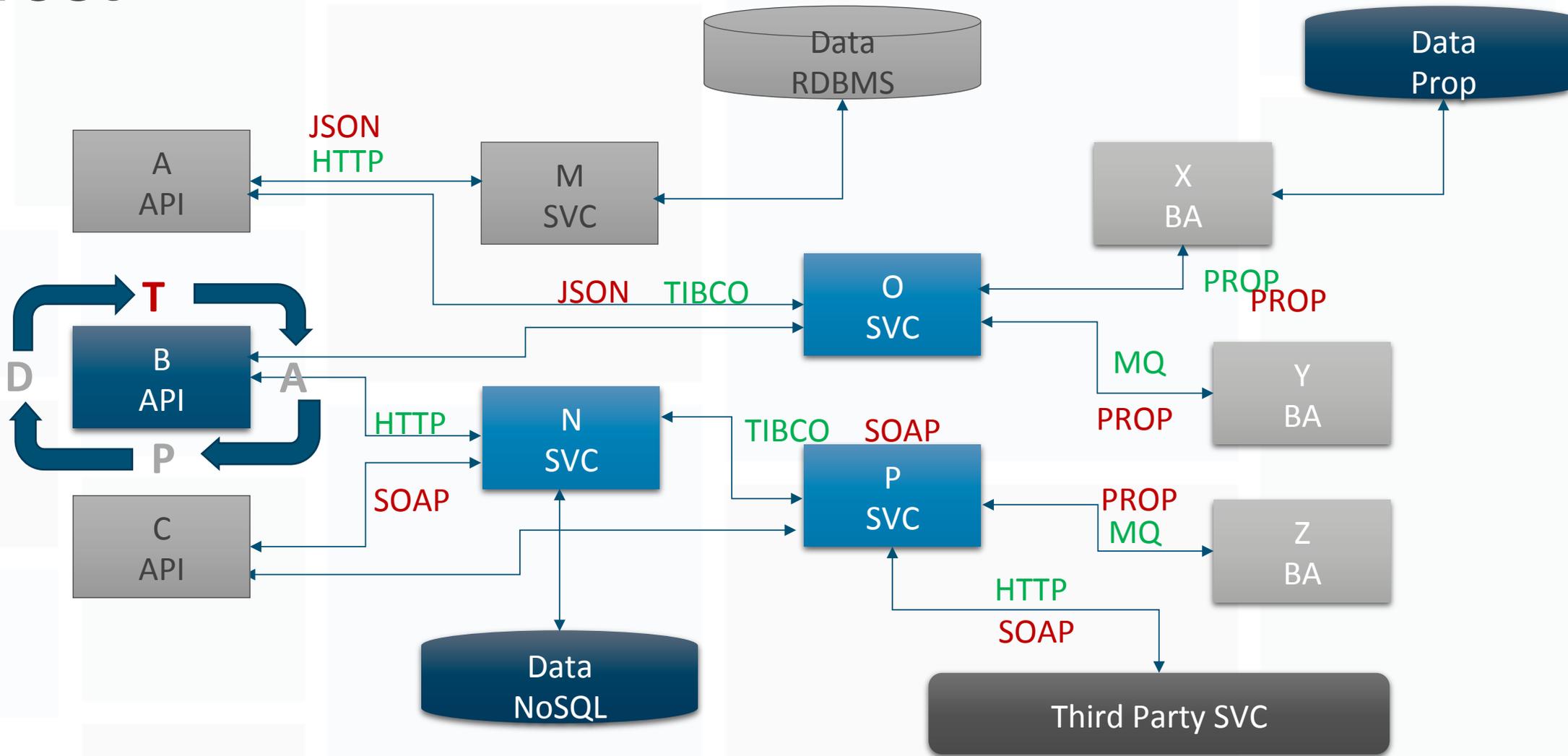


Development

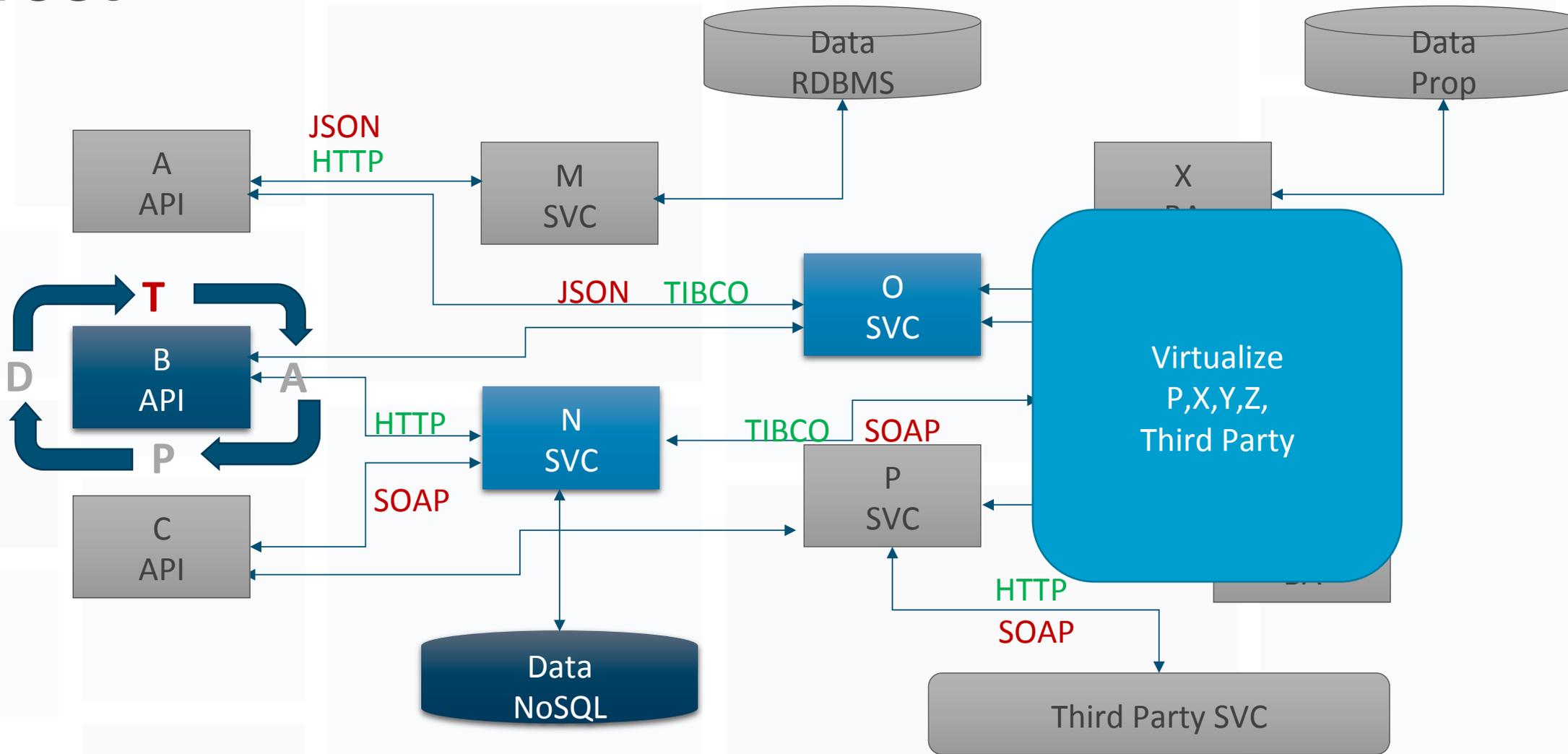


- Virtualize on local machine
- Unit test
- Functional tests
- Fully decoupled
- Fully independent

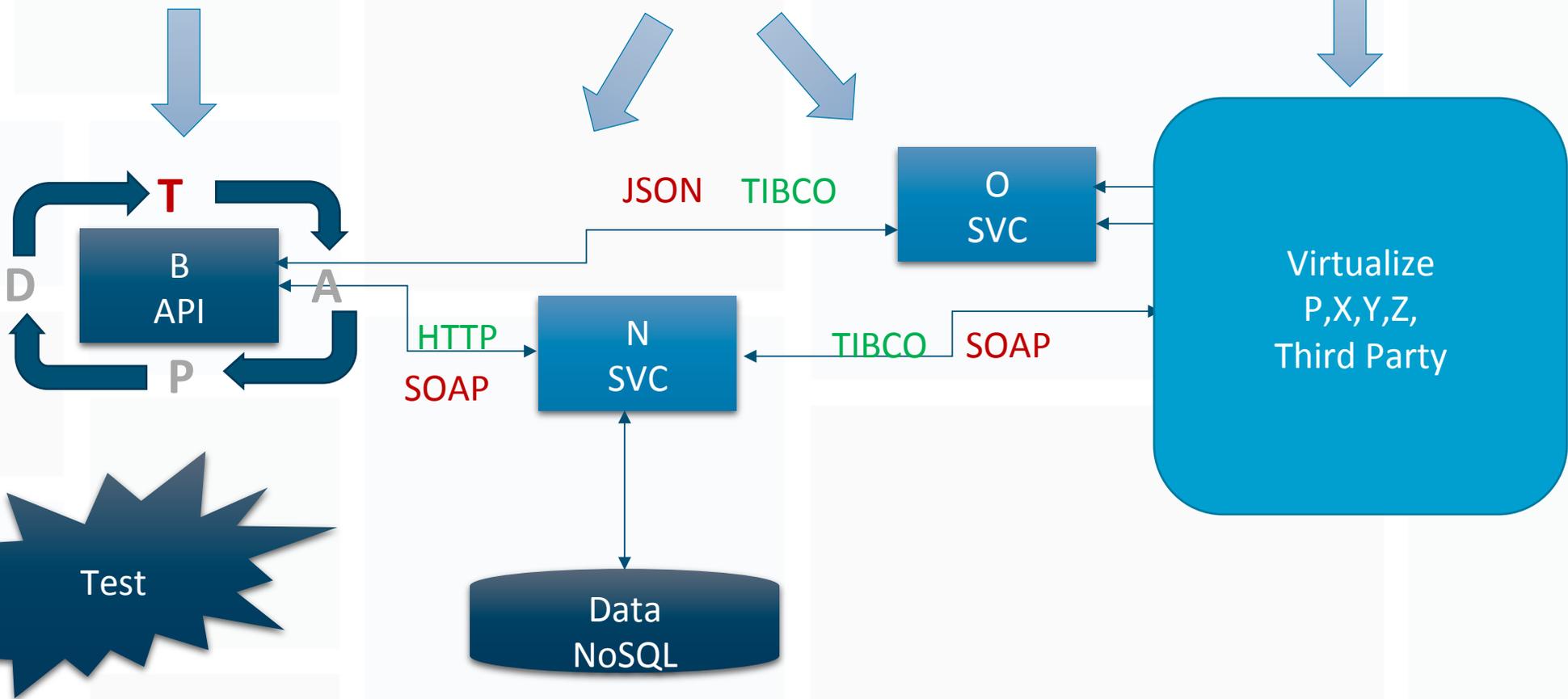
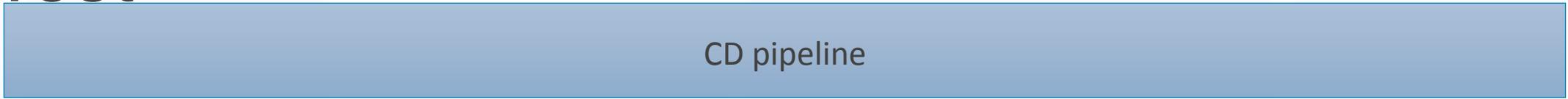
Test



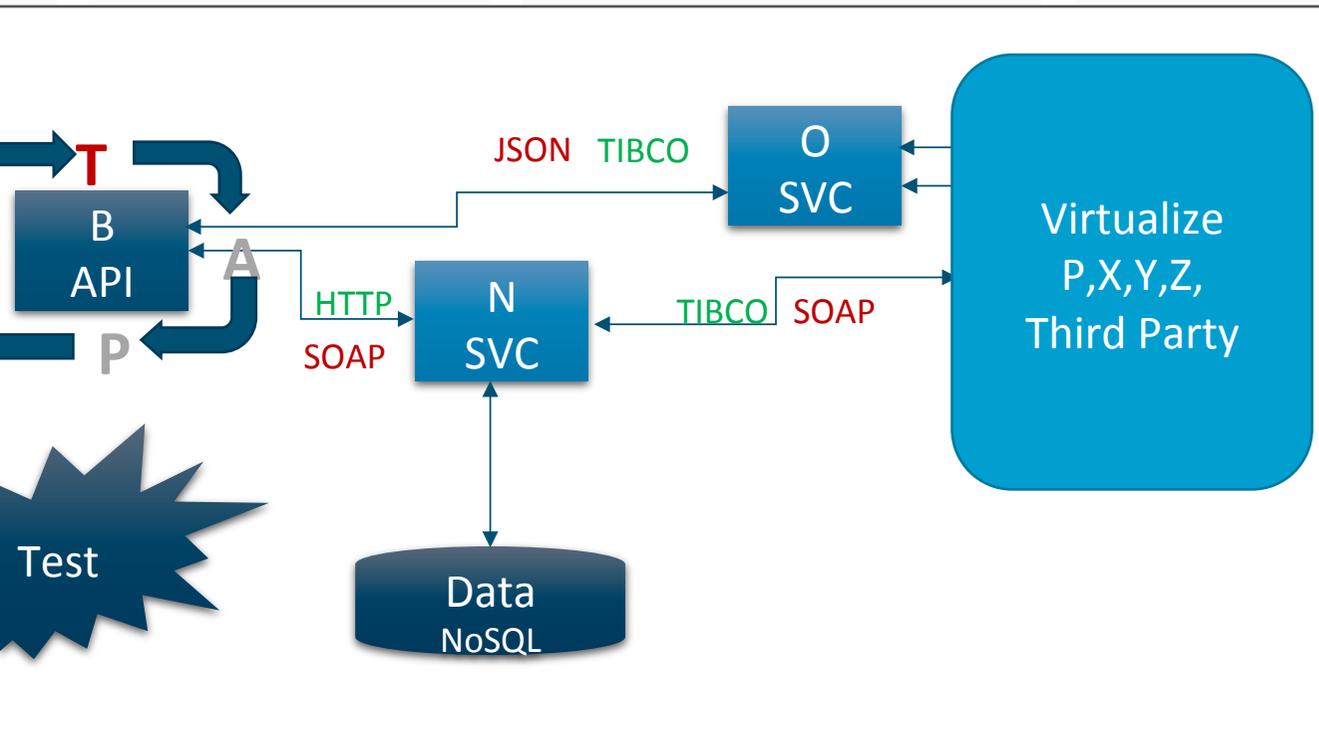
Test



Test



Test



- Choose “cut-off”
- Deploy AUT
- Deploy Neighbours
- Virtualize
 - On Server
 - Configured environment
 - Supports custom transport
 - Supports custom payload
- Functional tests
- Environment choice
 - Test chain

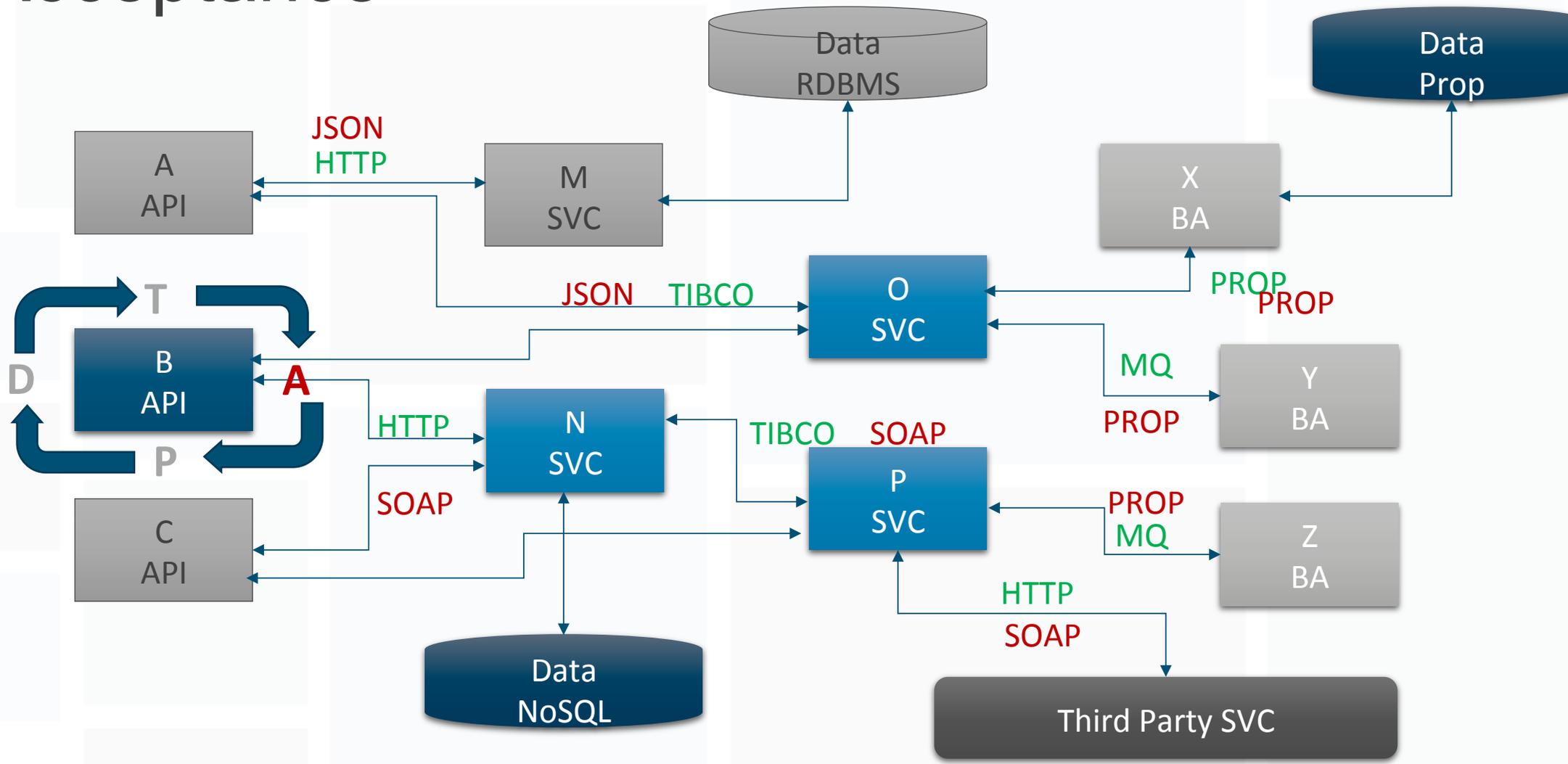
Acceptance

Focus on non-functional requirements

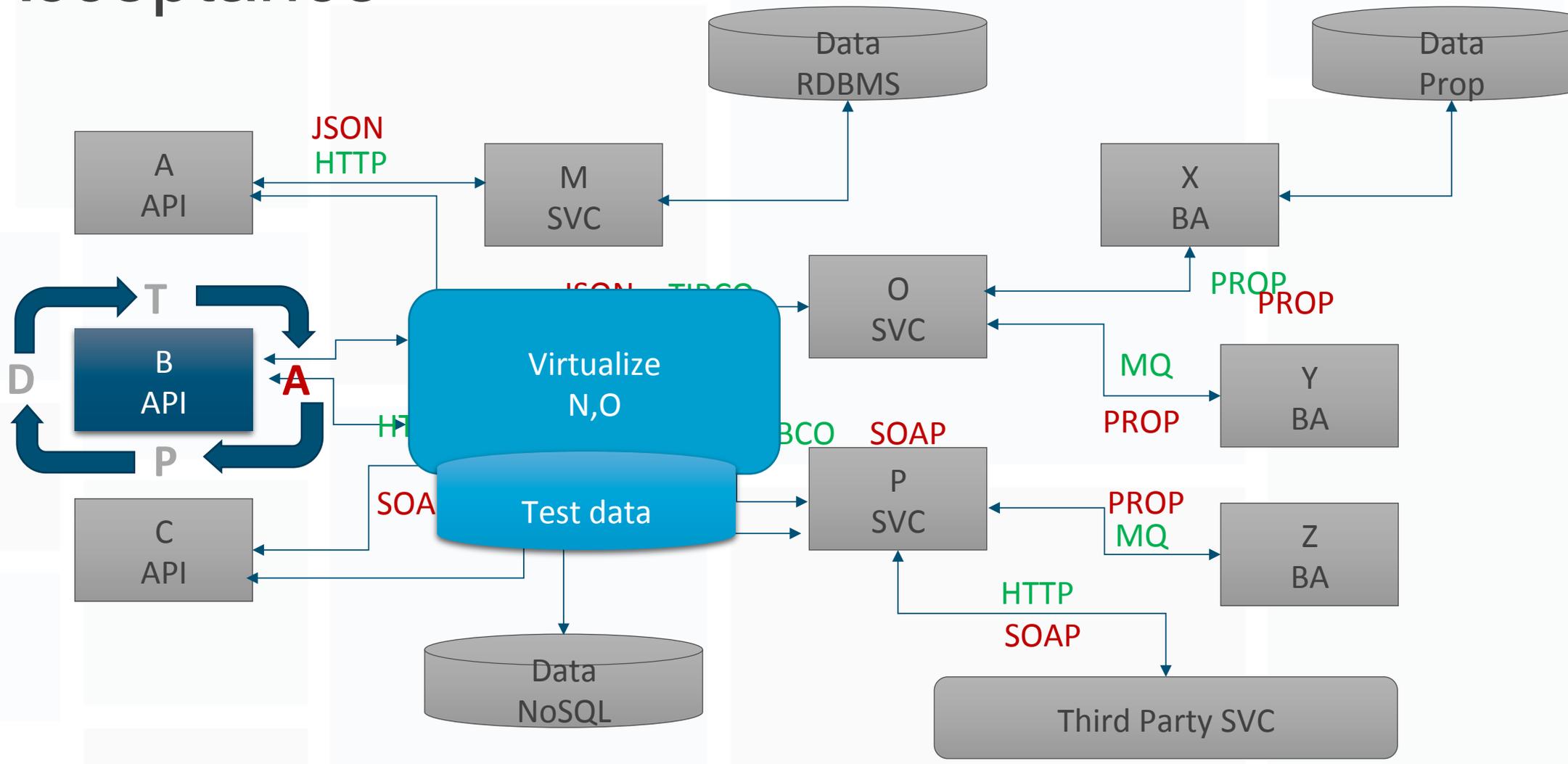
Component central

- DevOps team responsibility
- Capacity
- Stability
- Performance
- Resilience

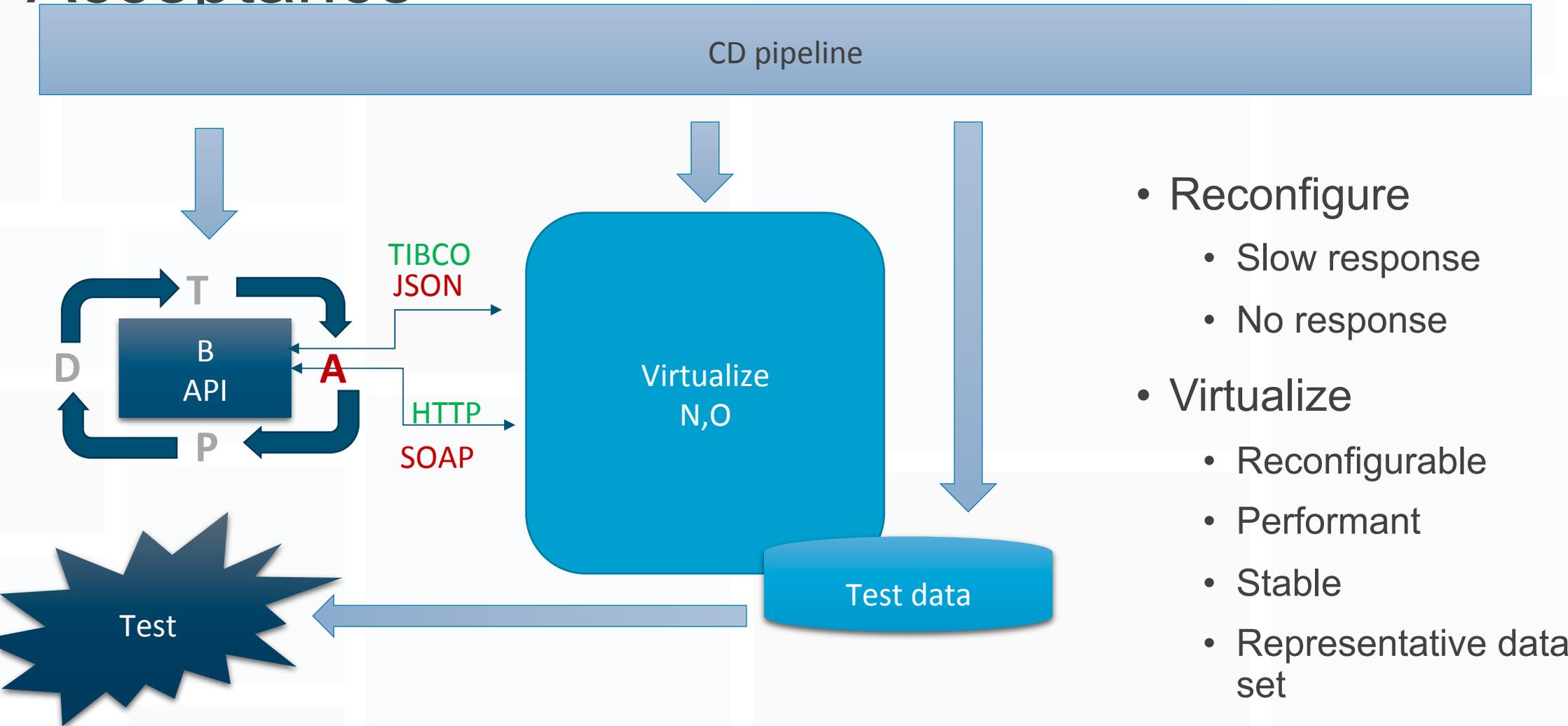
Acceptance



Acceptance



Acceptance



Pre production

Does not fit in pure DevOps

IMHO mandatory

AKA “staging”

Production like as possible

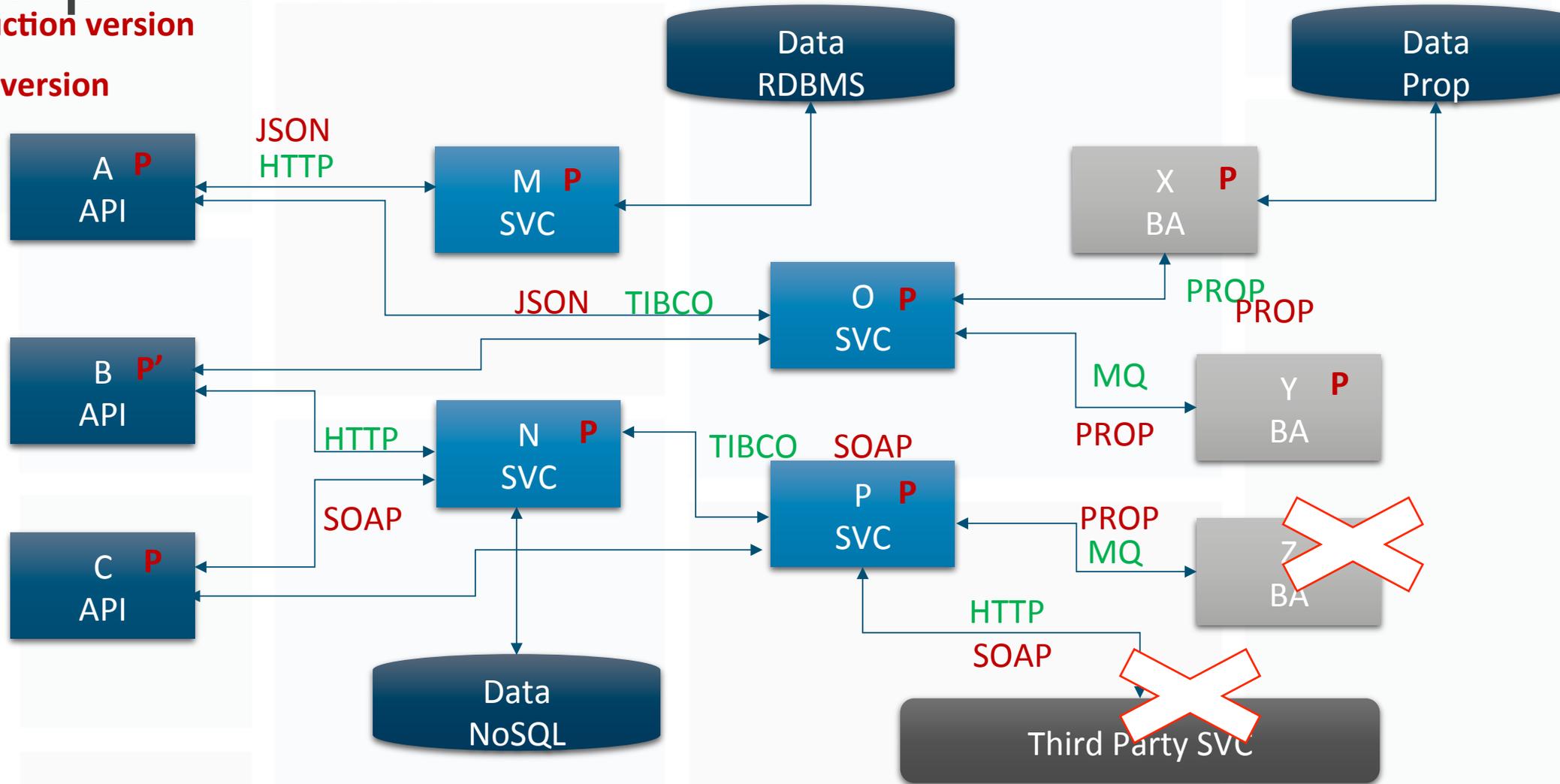
- Sizing
- Configuration
- Data

“Tomorrows production”

Pre-production

P: Production version

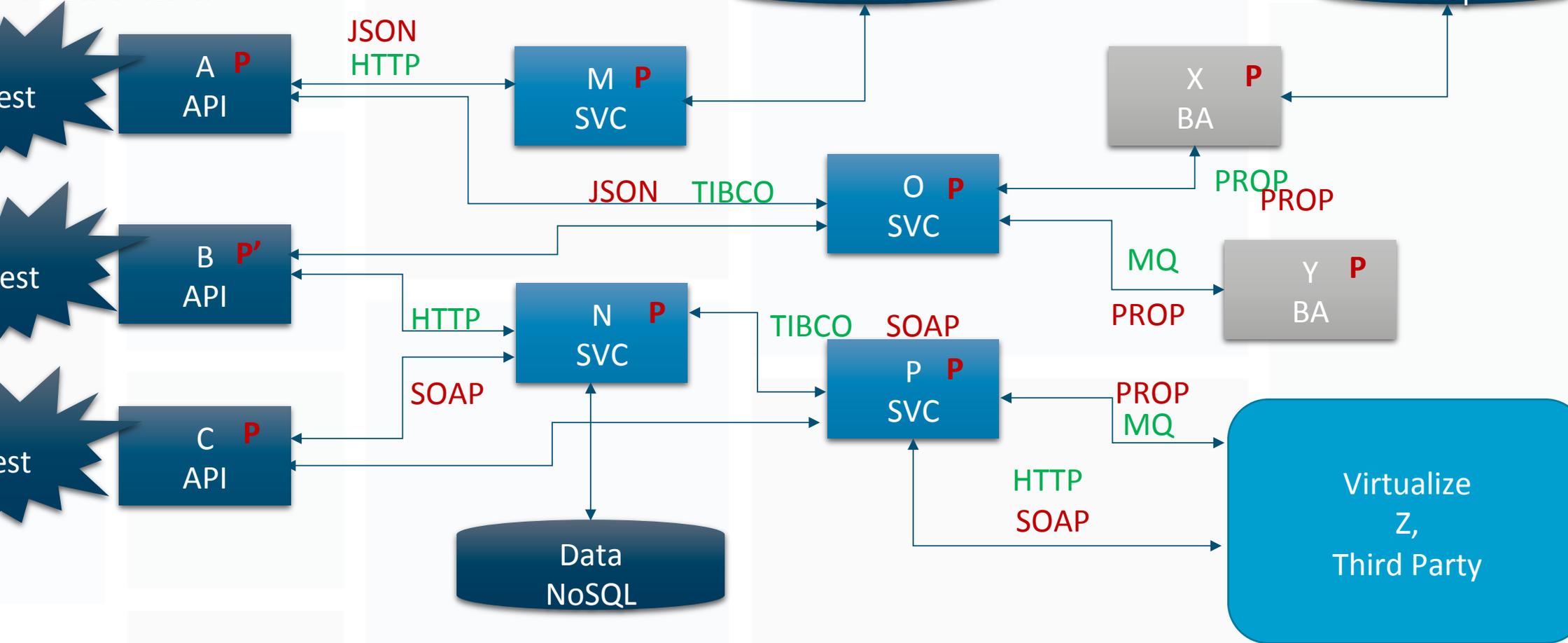
P': New version



Pre-production

P: Production version

P': New version



Agenda

Modern times and need for Virtualization

Virtualization per phase

- Development
- Test
- Acceptance
- Pre-production

Requirements & implementation

- Functionality (Dev)
- Deployment (Ops)

Wrap up

Deployment strategies

Two main deployment strategies:

Central: Client / Server model

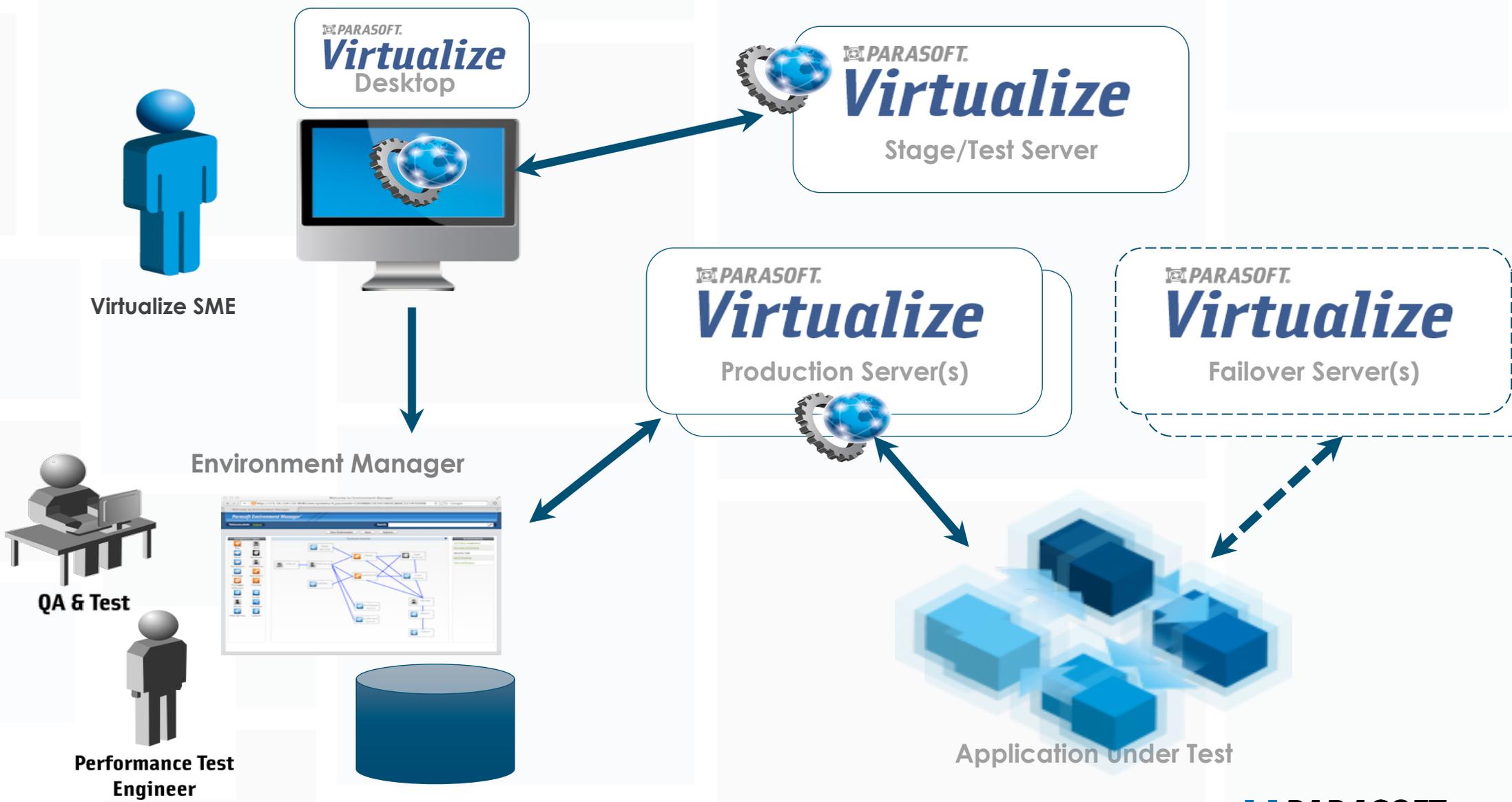
- Repository containing Implementations (PVA-files)
- Environment Manager to control provisioning
- Optional Load Balancing with multiple servers

De-central: Container or Cloud model

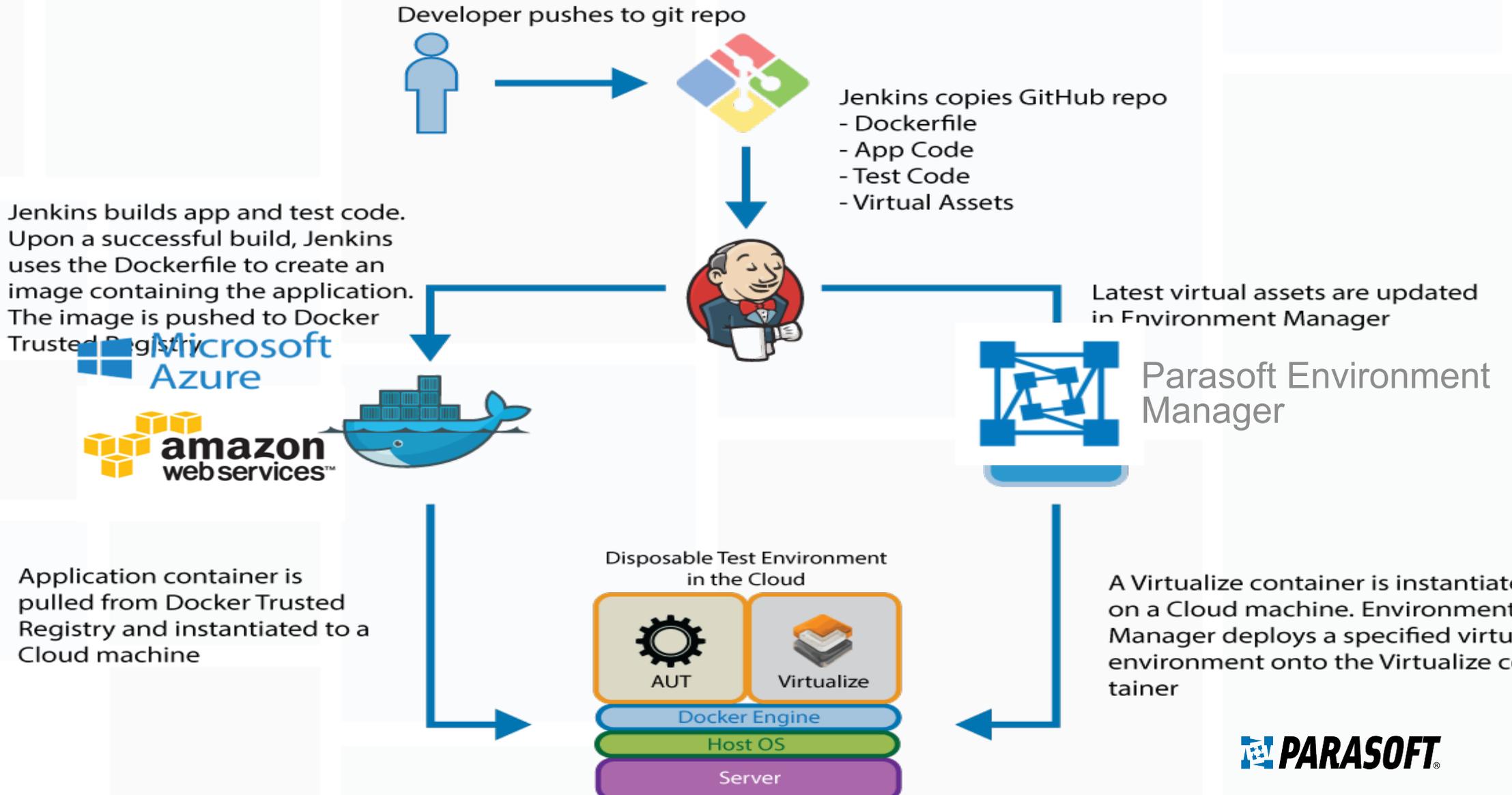
- API driven launch of local servers
- GIT based deployment of Implementation (PVA files)

Towards completely scripted construction

Central Deployment (client / server)



De-central Deployment (container based)



Towards scripted deployment

Docker & Java & Tomcat

Set protocol jars & Deploy Virtualize (war)

Static implementation:

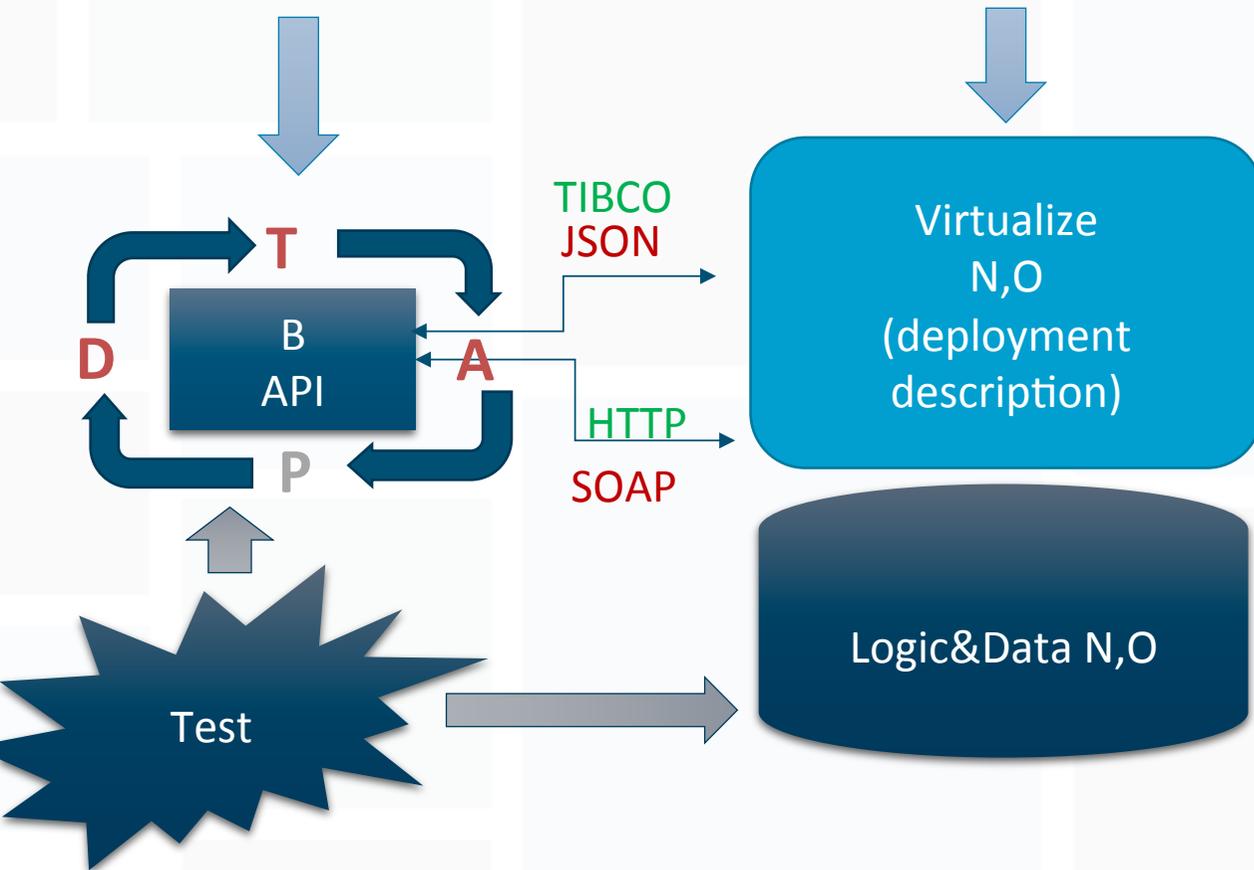
- Deploy PVAs (model) from GIT
- Provision test-data

Dynamic implementation:

- Build PVA files on the fly (using API)
- Example:
 - *Given “customer with 2 credit cards”*
 - *When “deleting one card”*
 - *Then “one card should be left”*

Towards scripted deployment

CD pipeline



- Pipeline:

- Deploy SUT (eg B API)
- Deploy Virtualize (docker)
- Initialize N,O placeholders

- Test-execution:

- Populate N, O (“Given”)
- Execute (“when/then”)
- (and repeat)

Deployment summary

Central server and central deployment:

- “Waterfall”: Central “test data team”

Central server and de-central deployment:

- “Agile”: teams have control over Virtual Assets

De-central Server and central deployment

- For 3rd party interfaces / complex backends

De-central server and de-central deployment

- “DevOps”: Deploy Virtual Assets as part of pipeline
 1. “library of PVA’s” and provisioning
 2. “on the fly creation” (given-when-then)

Agenda

Modern times and need for Virtualization

Virtualization per phase

- Development
- Test
- Acceptance
- Pre-production

Requirements & implementation

- Functionality (Dev)
- Deployment (Ops)

Wrap up

Wrap up

Virtualization decouples your components

Decoupling oils the DevOps / CD pipeline

Virtualization requires:

- Various deployment models
 - Local, Server, Docker, Cloud
- Ready to use transports and payloads
- Extensibility where needed
- Dynamic reconfigurable (provisioning)
- Scalable to use in performance test
- Integration in CD pipeline (APIs)

Demo & Do-it Yourself

Choose the version that's right for you.

NEW	SINGLE USER	SMALL TEAMS	SMALL TEAMS	ENTERPRISE	ENTERPRISE
	Virtualize Community Edition	Virtualize and SOAtest Professional Desktop	Virtualize On-Demand Server for Microsoft Environments	Virtualize and SOAtest Runtime Server	Virtualize Performance Server
	Free	\$1,250/year Annual subscription	\$5.95*/hour Free 30-day Trial	Contact Parasoft for pricing	Contact Parasoft for pricing
	Single User Desktop	Machine-Locked Desktop <small>(Contact Parasoft for perpetual licensing options)</small>	On-Demand Cloud Server + Thin-Client	Desktop + Server + Thin-Client	Desktop + Server + Thin-Client
	CAPABILITIES Service Virtualization	CAPABILITIES Service Virtualization API and Web Testing	CAPABILITIES Service Virtualization Test Environment Management Test Data Management	CAPABILITIES Service Virtualization API, Web, and Load Testing Test Environment Management Test Data Management	CAPABILITIES Service Virtualization API, Web, and Load Testing Test Environment Management Test Data Management
	PROTOCOLS HTTP, HTTPS	PROTOCOLS HTTP, HTTPS, HTTP/2 JMS, JDBC MQ, MQTT AMQP RabbitMQ	PROTOCOLS HTTP, HTTPS	PROTOCOLS HTTP, HTTPS, HTTP/2 JMS, JDBC MQ, MQTT AMQP RabbitMQ	PROTOCOLS HTTP, HTTPS, HTTP/2 JMS, JDBC MQ, MQTT AMQP RabbitMQ

Azure marketplace

Select a software plan

Service Virtualization On-Demand

Usage-based monthly billed SKU

Starting at

US\$5.95/hour



Pricing by virtual machine instance

[Download table as CSV](#)

Show: Publisher recommendations All virtual machine instances

Region

Central US



The publisher recommends the following 6 virtual machine instances for use with this software plan.

Virtual Machine		Configuration				Cost per hour		Total cost	
Instance	Category	Cores	RAM	Disk Space	Drive Type	Infrastructure Cost	Software Cost	Hourly	Monthly
A2	General Purpose	2	3.5GB	60GB	HDD	US\$0.085	US\$5.95	US\$6.035	US\$4,490.04
A3	General Purpose	4	7GB	120GB	HDD	US\$0.188	US\$5.95	US\$6.138	US\$4,566.672
DS2*	General Purpose	2	7GB	100GB	SSD	US\$0.154	US\$5.95	US\$6.104	US\$4,541.376
DS3*	General Purpose	4	14GB	200GB	SSD	US\$0.308	US\$5.95	US\$6.258	US\$4,655.952

Visual Studio

Parasoft Virtualize / SOAtest Desktop

for Microsoft Visual Studio Enterprise Subscriptions



Visual Studio Enterprise subscribers benefit from 6 months for FREE followed by 25% off the annual subscription price

Go to your [Visual Studio account](#) to activate your free subscription



The Visual Studio Blog

The official source of product insight from the Visual Studio Engineering Team

New benefits for Visual Studio subscribers and Dev Essentials members

March 7, 2017 by [Visual Studio Blog](#) // 0 Comments



Today at the [Visual Studio 2017 launch event](#) we announced a set of new and updated benefits for our subscribers. If you missed the event or want to watch the on-demand trainings, check out the [launch event page](#). If you're a current Visual Studio subscriber or Dev Essentials program member [activate your new benefits](#) to get started right away. To learn more about our developer subscriptions and benefits, visit the [Visual Studio site](#).

We're pleased to announce the following new and updated benefits:

Office365 Dev Account

Visual Studio Enterprise subscribers now get multi-user access with 25-seat Office365 developer account to create and test applications for Office365.

Opsgility on-demand training

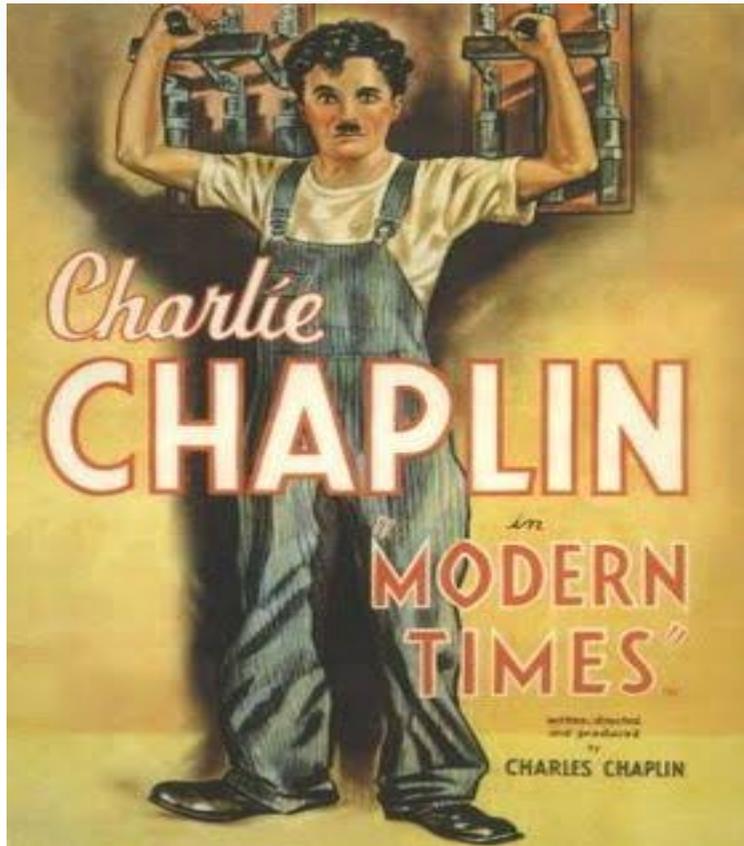
Visual Studio Dev Essentials members now have 3-month subscription to one of the most comprehensive video libraries on Azure curriculum available. After the 3-month period, members will be eligible to upgrade to a full Opsgility membership if they opt in, including full lab guides and more. For more information, see [Opsgility's blog](#).

Parasoft Virtualize/SOAtest Professional Desktop

Visual Studio Enterprise subscribers get a 6-month subscription with full access to Parasoft Virtualize/SOAtest Professional Desktop. These tools provide test automation to help ensure the security, reliability, and performance of transactions across modern applications and systems. Simulate and test the behavior of unavailable and evolving applications and systems. At the end of the 6-month subscription, subscribers will qualify for a 25% discount on the price of annual subscription. For more information, check out [Parasoft's benefit page](#).



Q & A



- Thank you