



Service Virtualization: Expanding the Breadth and Scope of Security Testing

Arthur Hicken – Parasoft
Evangelist / Security Specialist

ITEA - November 2013

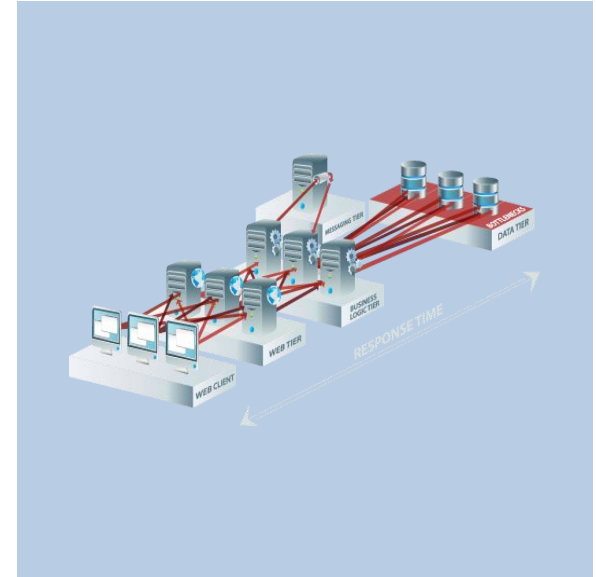
Major Disruptions In the SDLC



Cloud challenges ingrained concepts about security



Mobile devices force the industry to re-think the user experience



APIs drive composite apps and interconnecting multiple dependencies

What needs to be tested increasing – exponentially

- De-coupled and evolving system components (Composite / SOA)
- Larger, complex architectures
- Cloud – growth of external dependencies

Who is involved with software getting more complex

- Large, distributed teams
- Agile, iterative development changes expectations

Many moving parts make it difficult & expensive to “stage”

- Mobile Device
- Service Endpoints
- 3rd party systems

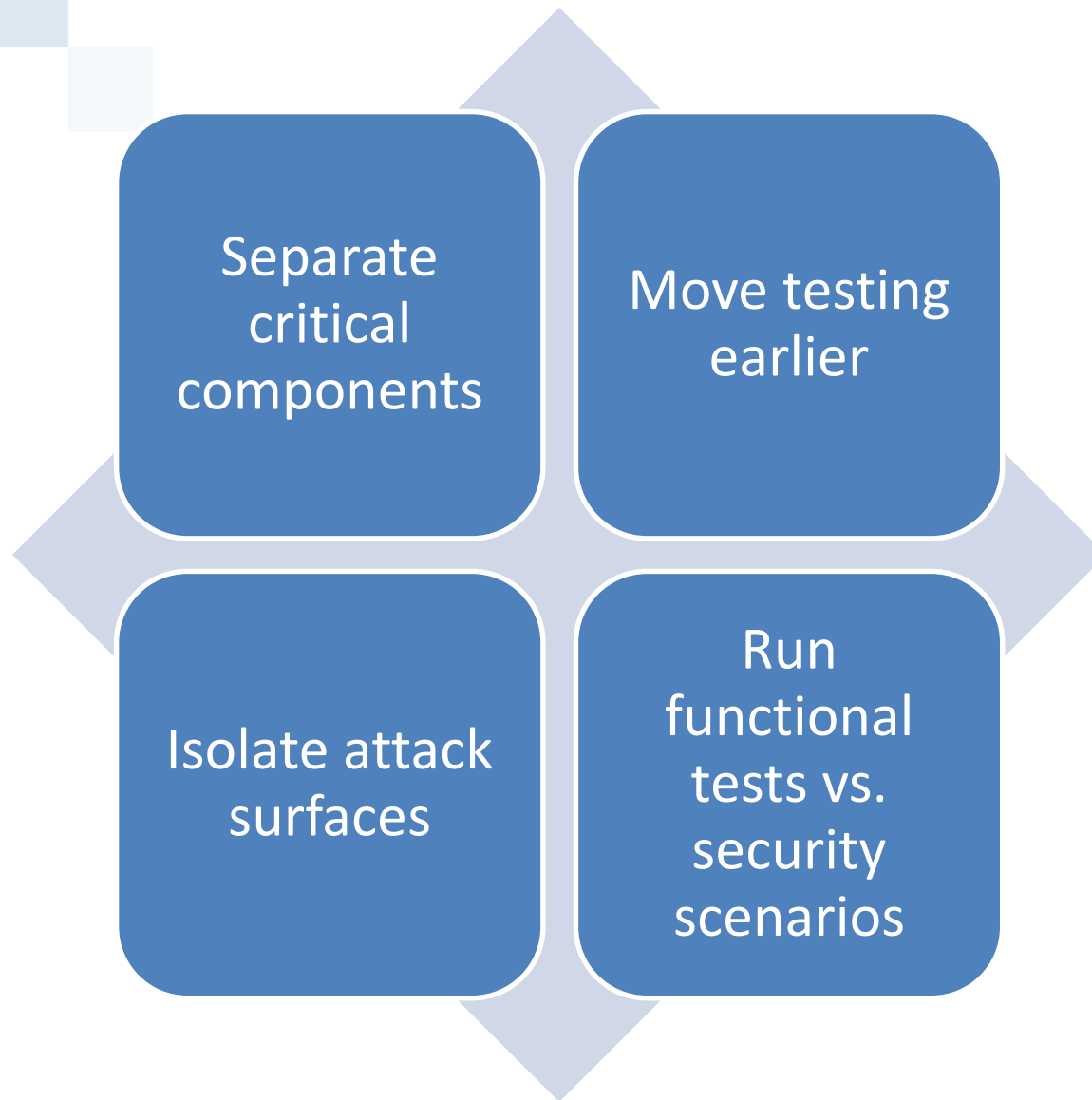
The average number of dependent application associated with the System Under Test (SUT)

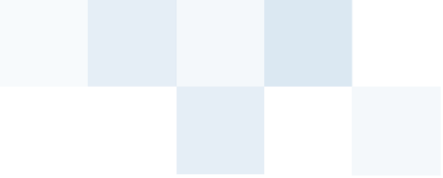
30

BUT, Dev/QA only have “trustworthy” access to **SIX** of the applications


1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30



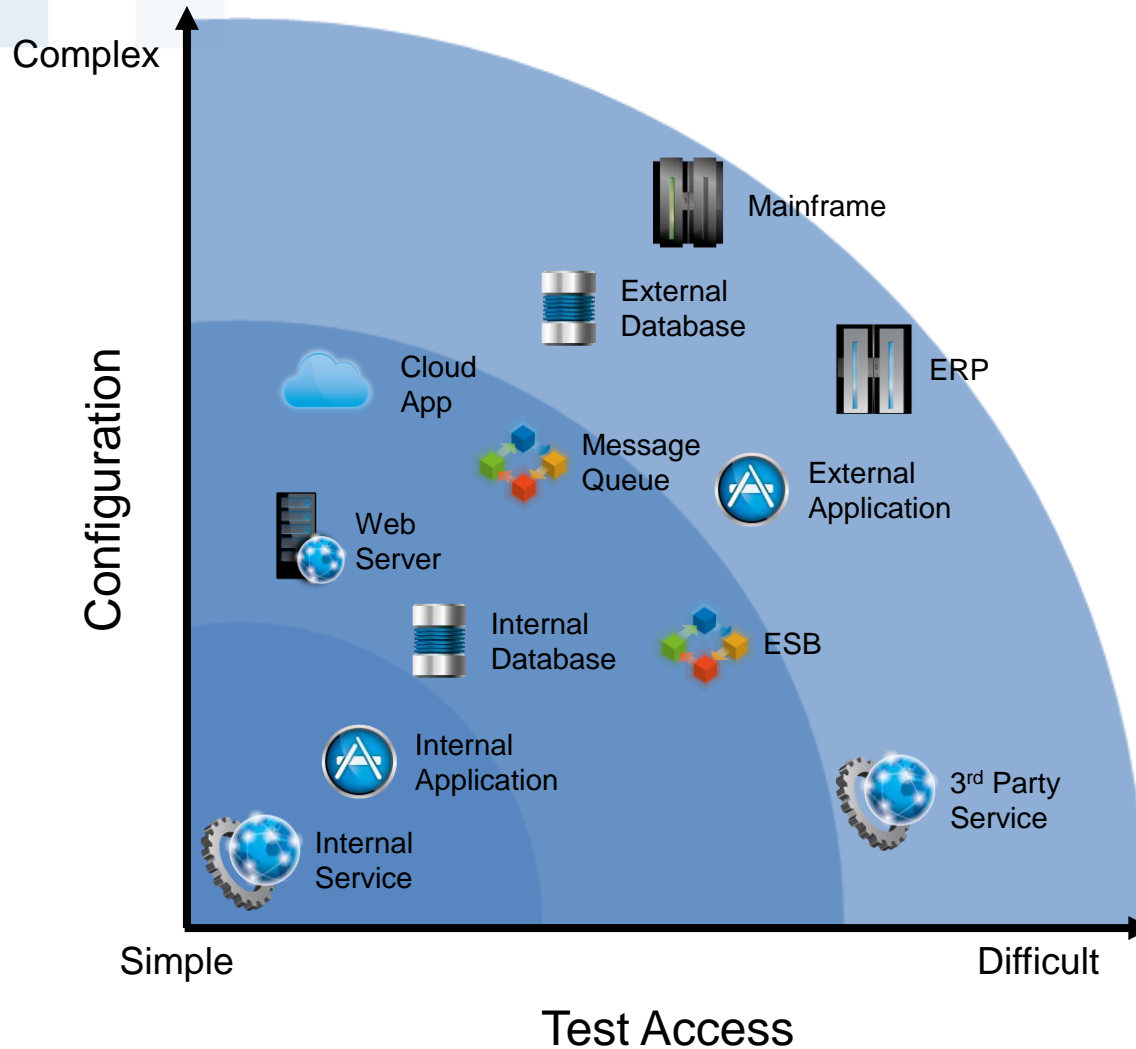




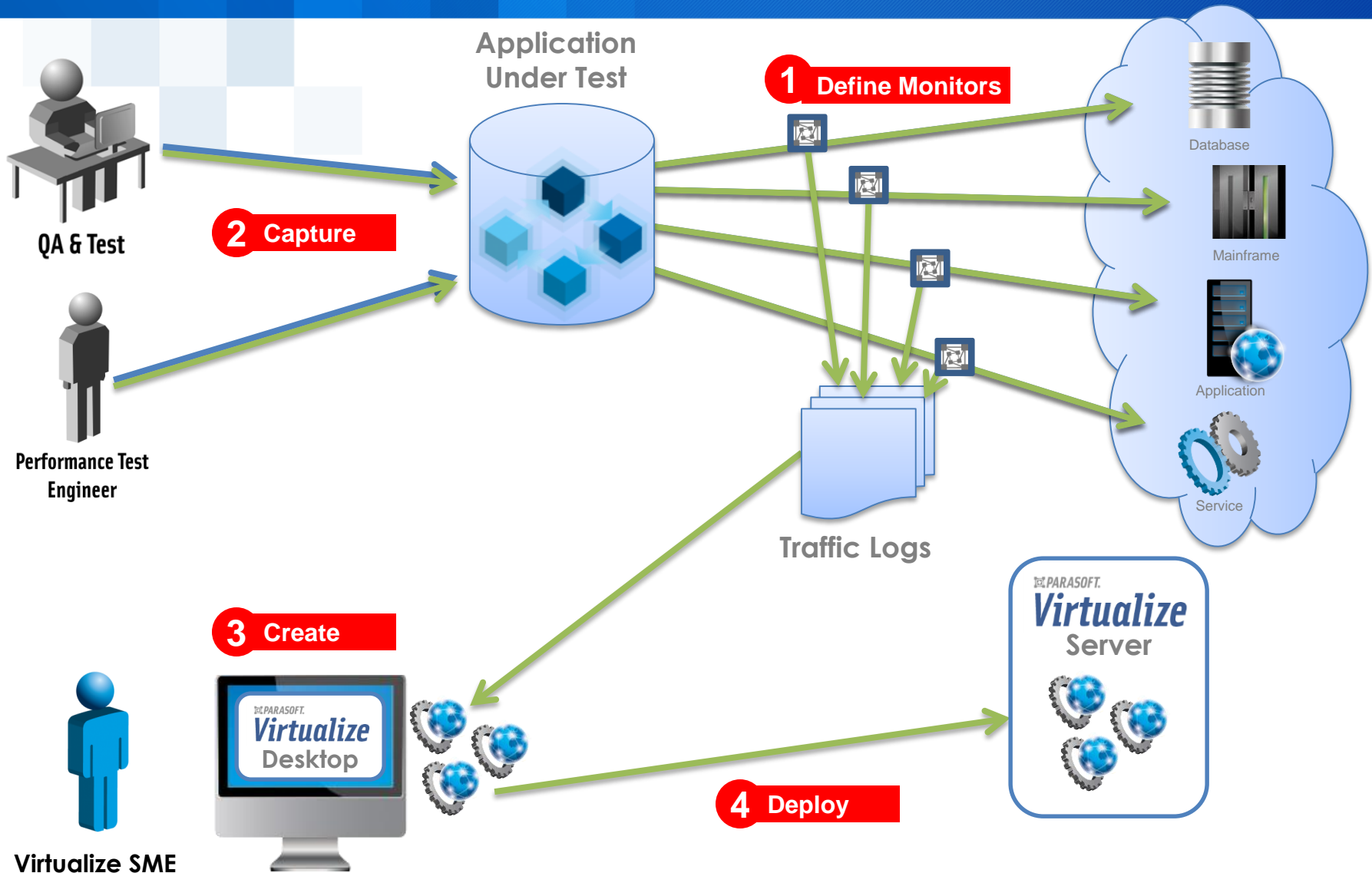
Service Virtualization
simulated dev / test environment
allowing you to test
anytime or anywhere



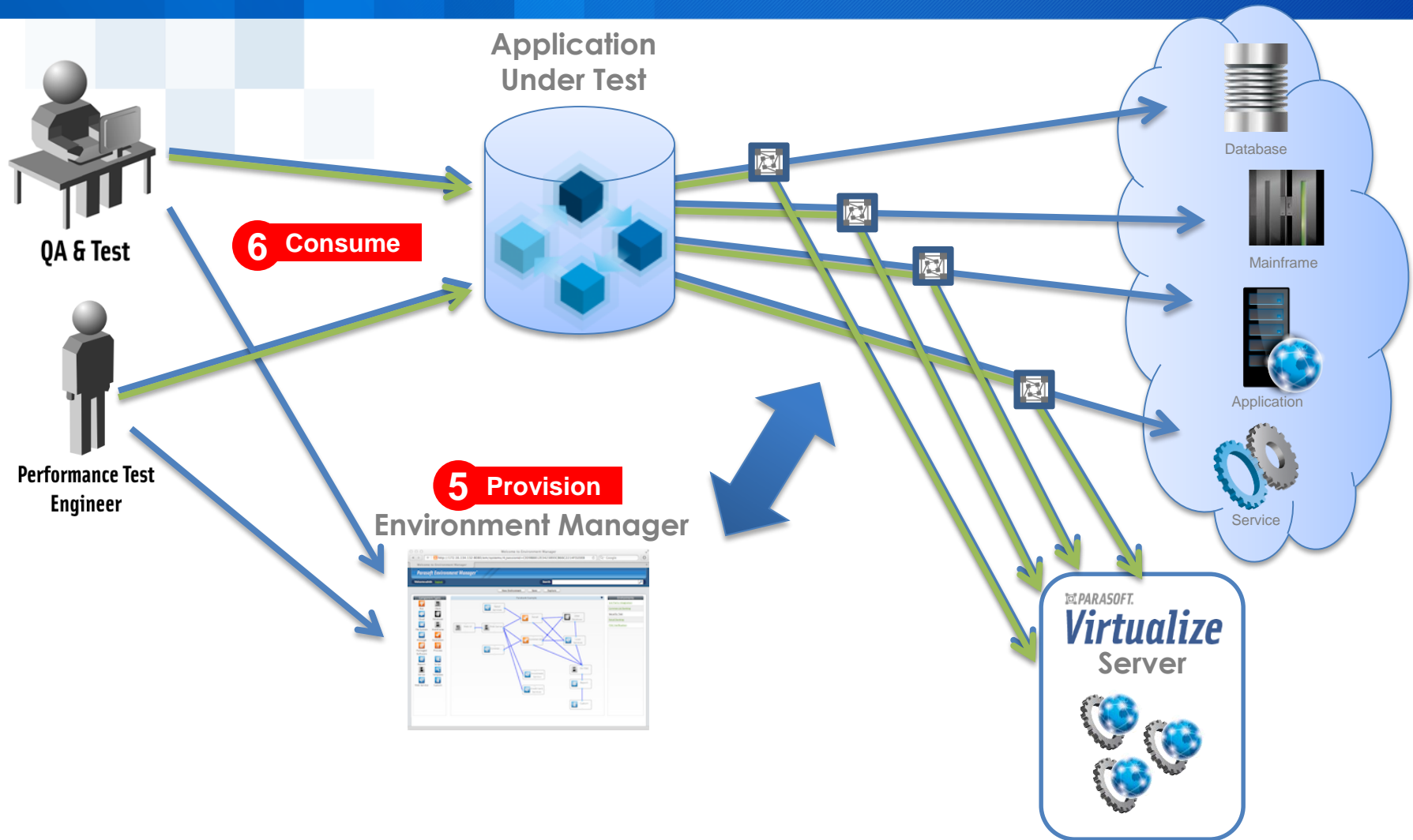
Test Environment Access



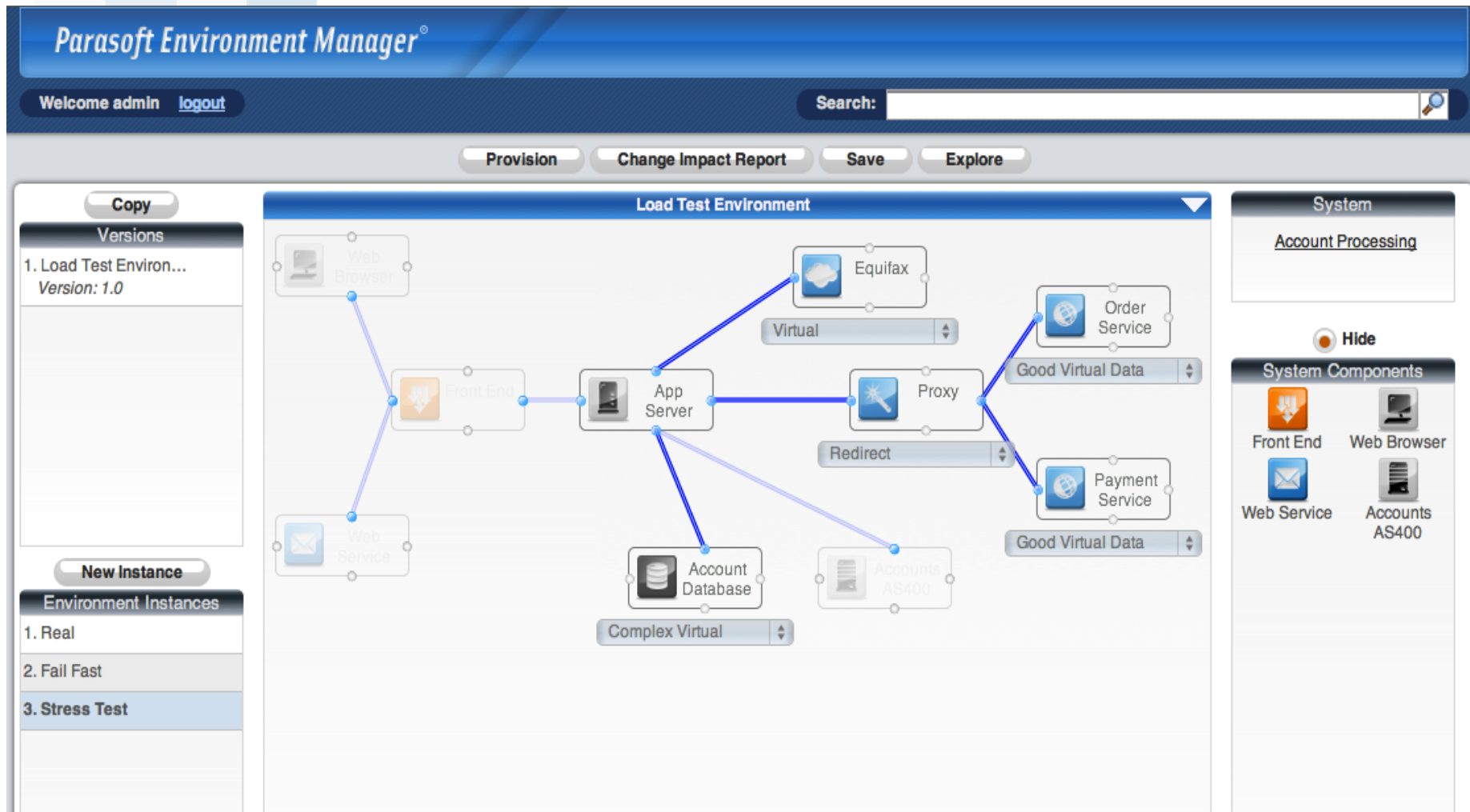
Parasoft Virtualize: How does it work?



Parasoft Virtualize: How does it work?



Rapid Environment Access



Service Virtualization provides a complete environment for developing and testing versus complex, dependent systems

Stubs

- Inside-out approach that disassociates a test case with a dependent systems
- Brittle
- Limited reuse
- Static response

Service Virtualization

- Creates an environment in which to run “rich” test scenarios
- Virtualized assets represent real system behavior
- All virtualized-assets are reusable
- Virtualized assets are programmable and extensible

Web applications and Web services:

- Injection flaws (including SQL injection)
- Unvalidated input
- Capture and replay
- Buffer overflows
- Denial of service
- Improper error handling
- Broken access control

Web applications

- Cross-site scripting
- Broken authentication and session management
- Insecure storage

Web services

- WSDL access and scanning
- XML external entity
- XML bombs
- Large payloads
- XPath injections



Easier to test end-to-end



Testing can begin before the software is finished



Components and sub-systems
can be tested in isolation

- Web

- <http://www.parasoft.com/jsp/resources>

- Blog

- <http://alm.parasoft.com>

- Social

- Facebook: <https://www.facebook.com/parasoftcorporation>

- Twitter: @Parasoft @MustRead4Dev @CodeCurmudgeon

- LinkedIn: <http://www.linkedin.com/company/parasoft>

- Google+ Community: Static Analysis for Fun and Profit

