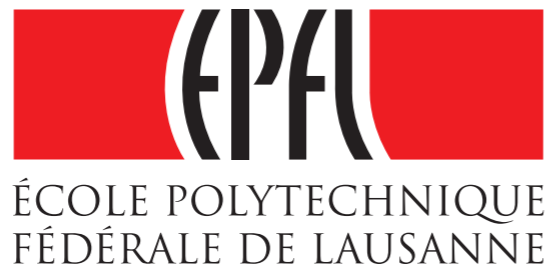


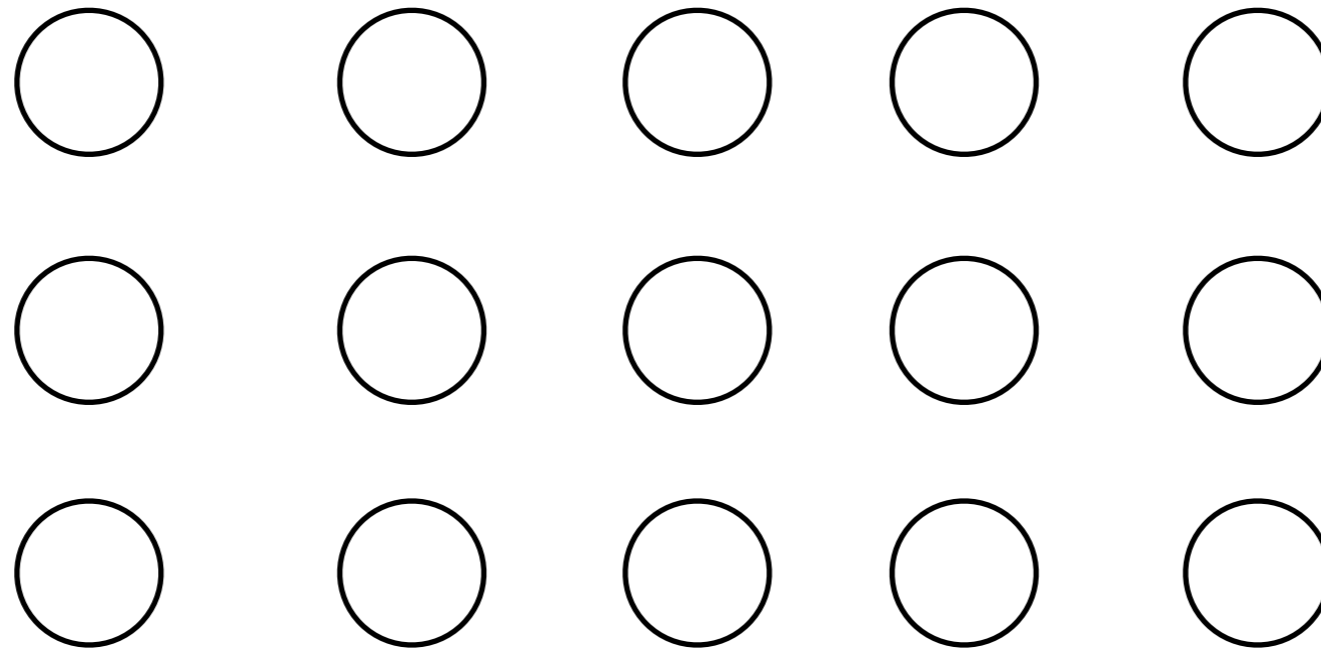
Staged Deployment in Mirage, an Integrated Software Upgrade Testing and Distribution System

Olivier Crameri, Nikola Knezevic, Dejan Kostic,
Ricardo Bianchini, Willy Zwaenepoel

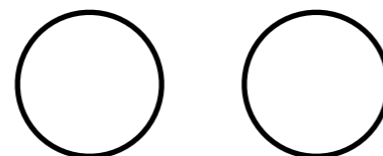


Software upgrade deployment

OUTSIDE
WORLD

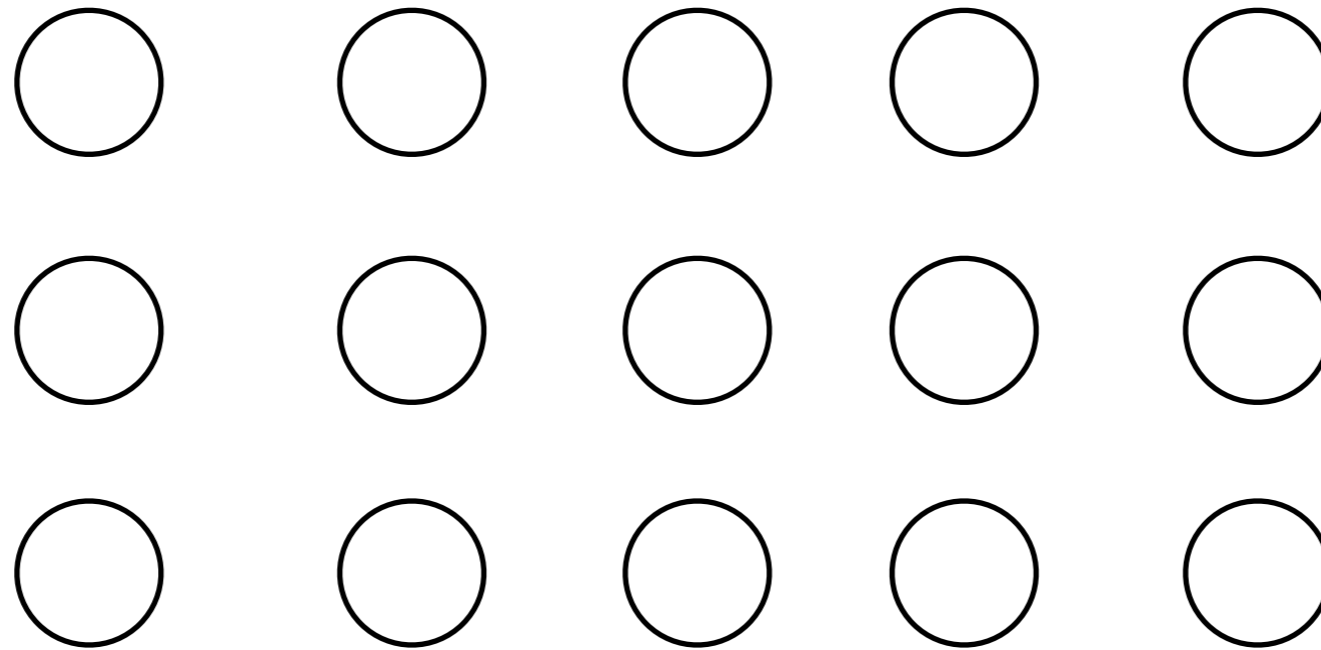


VENDOR

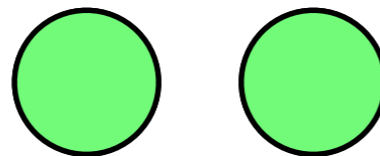


Software upgrade deployment

OUTSIDE
WORLD

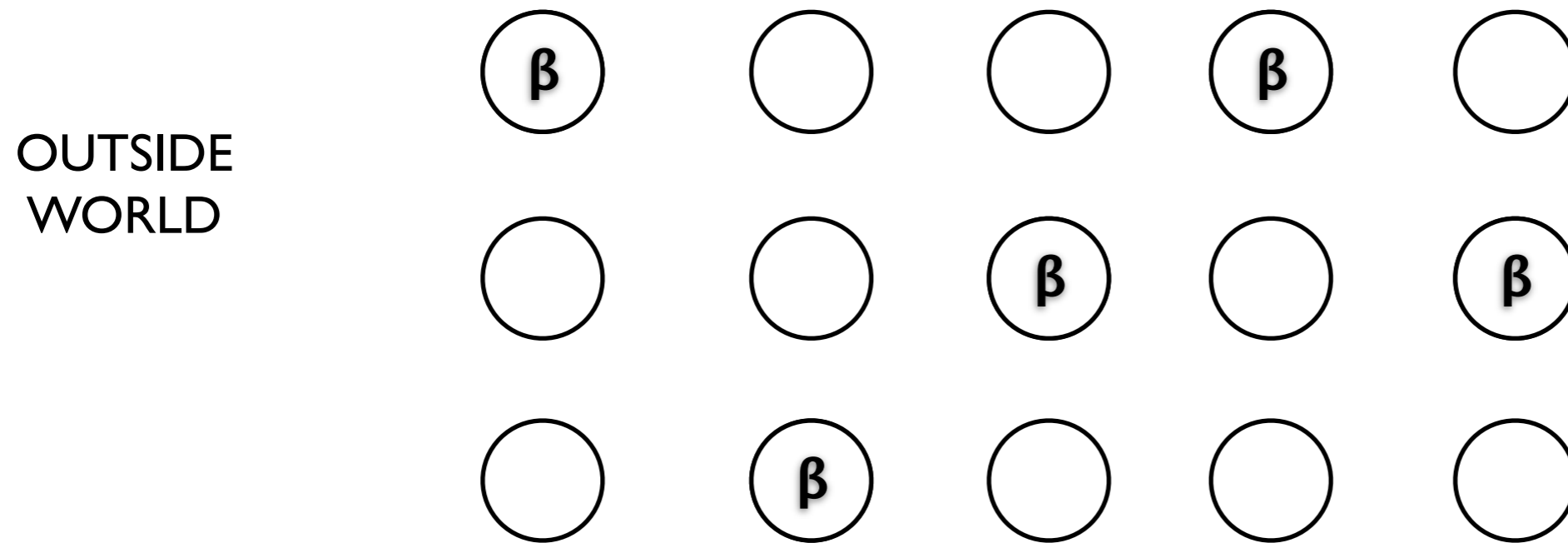


VENDOR

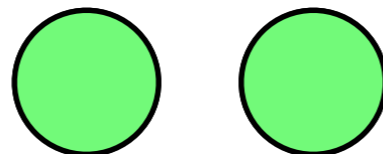


Software upgrade deployment

β : Beta-testers

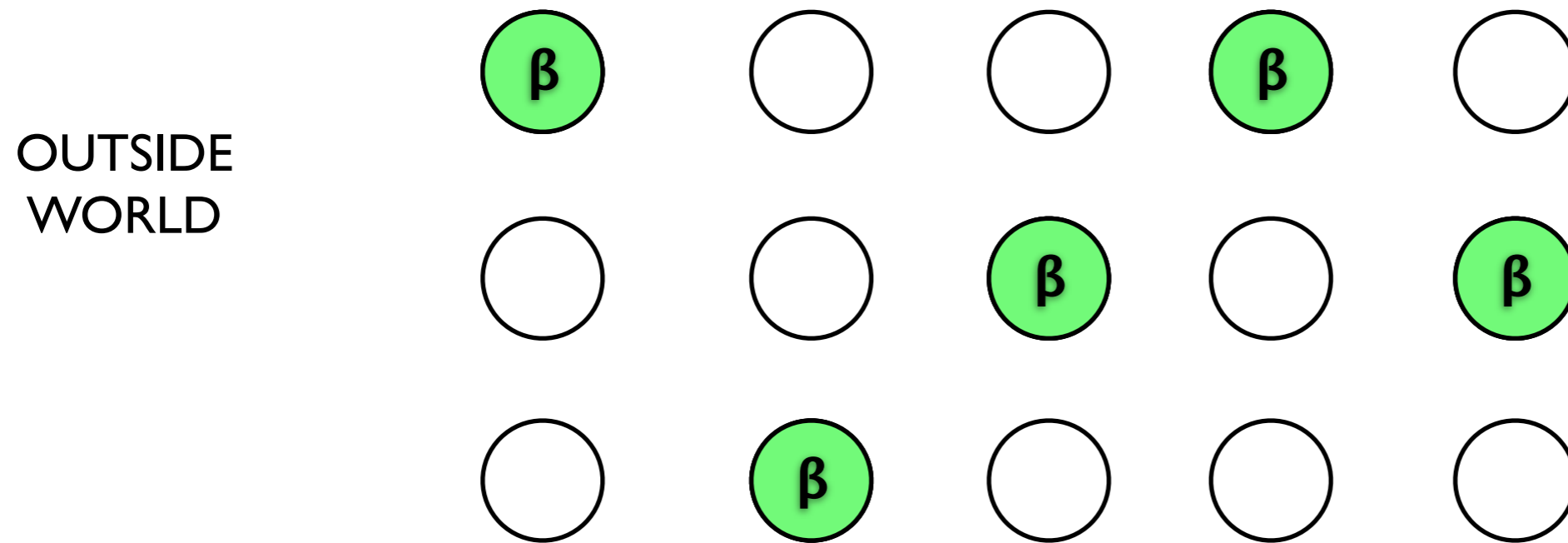


VENDOR

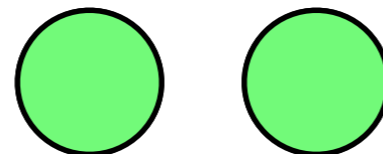


Software upgrade deployment

β : Beta-testers

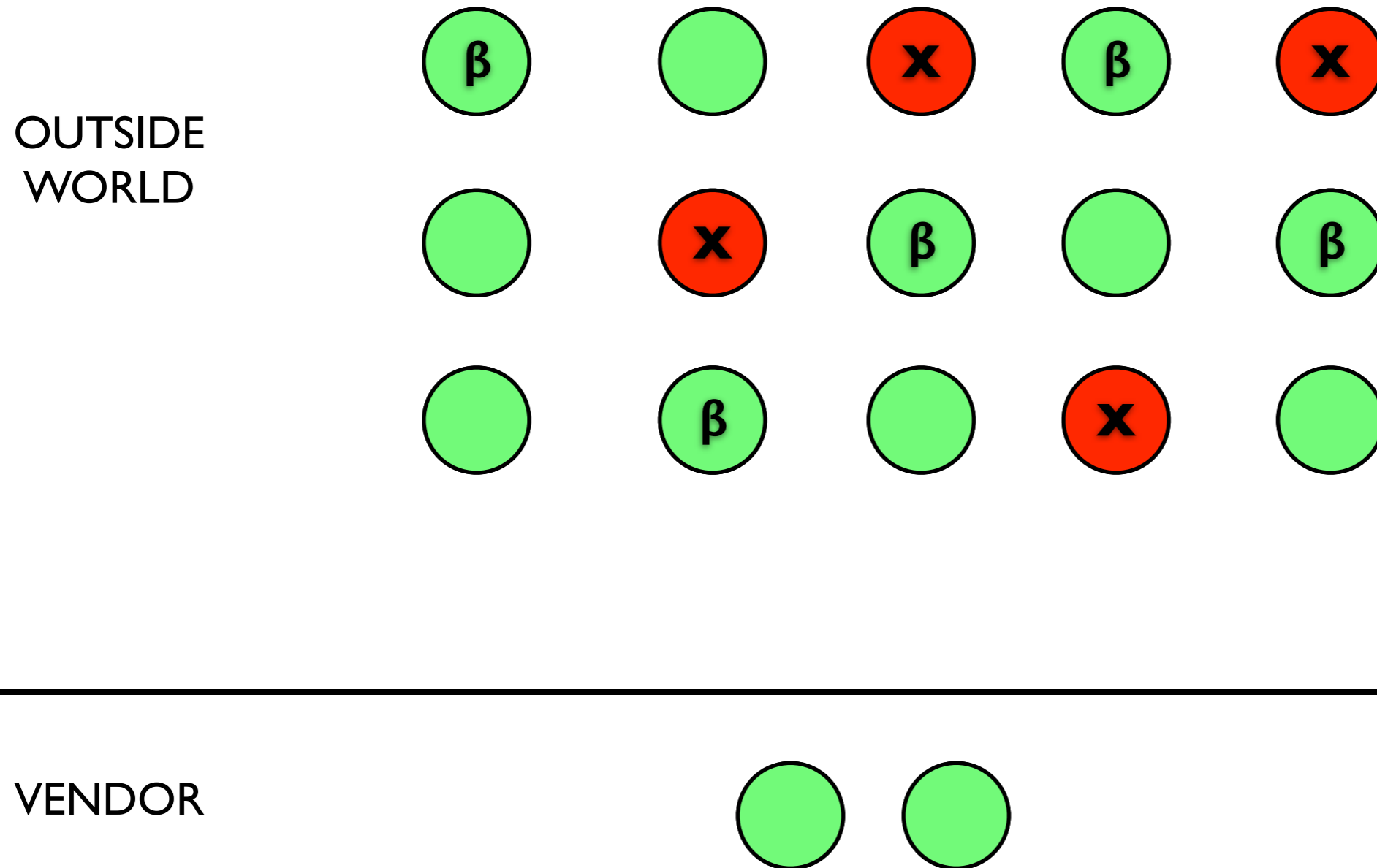


VENDOR



Software upgrade deployment

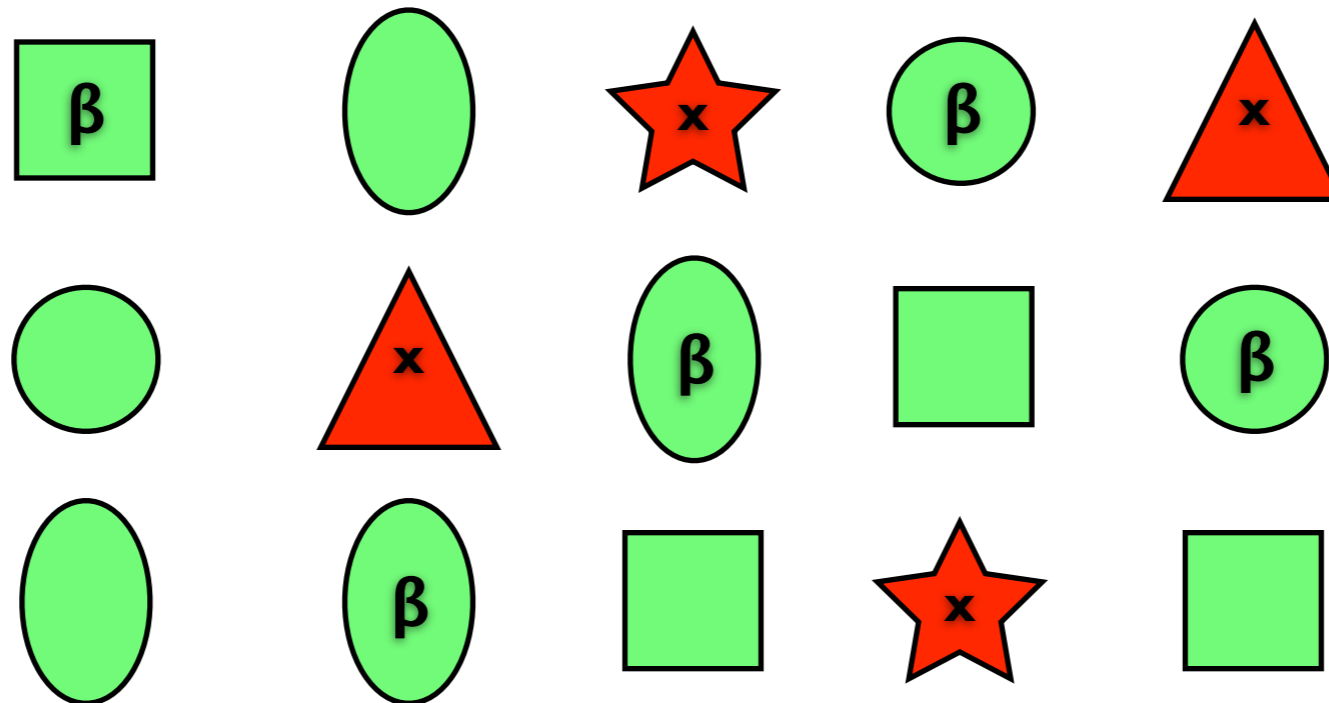
β : Beta-testers



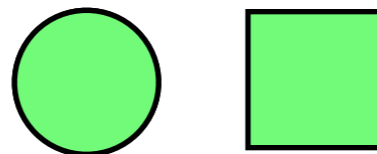
Software upgrade deployment

β : Beta-testers

OUTSIDE
WORLD



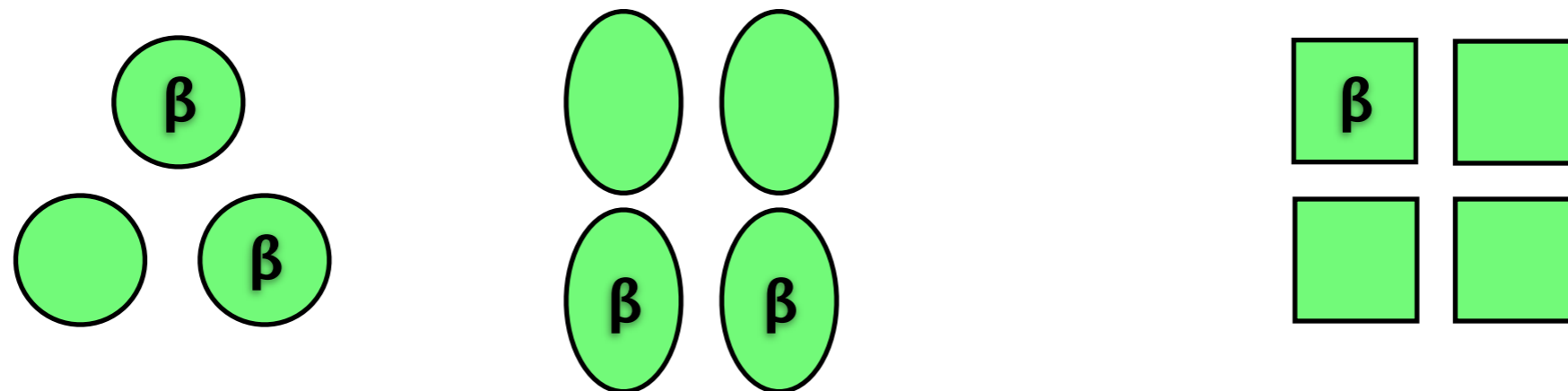
VENDOR



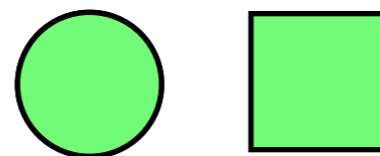
Software upgrade deployment

β : Beta-testers

OUTSIDE
WORLD

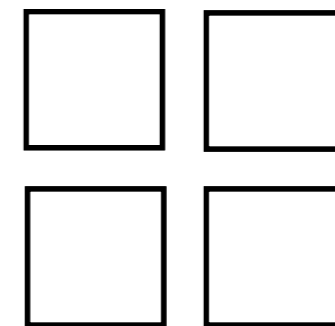
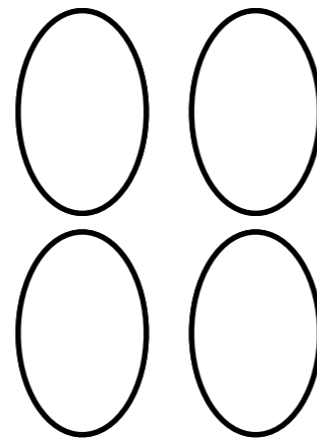
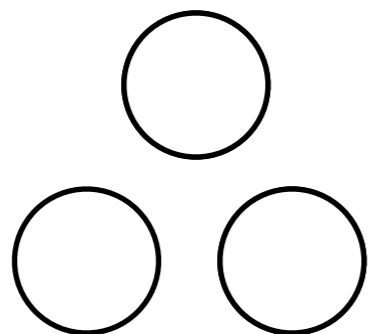


VENDOR



Staged deployment in Mirage

OUTSIDE
WORLD

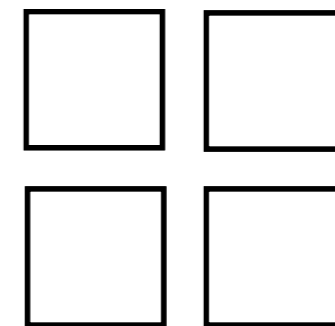
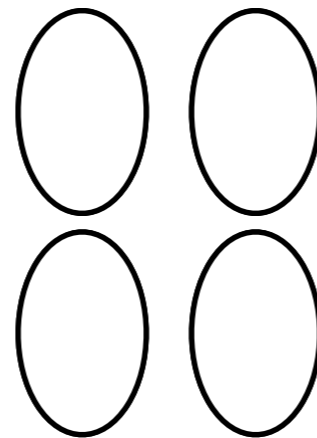
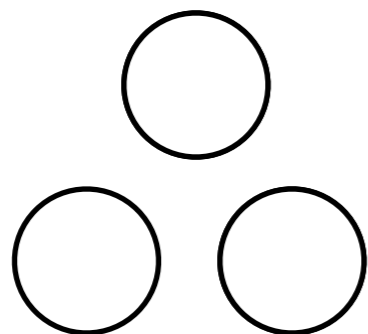


VENDOR

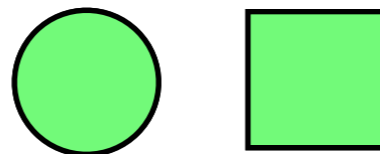


Staged deployment in Mirage

OUTSIDE
WORLD



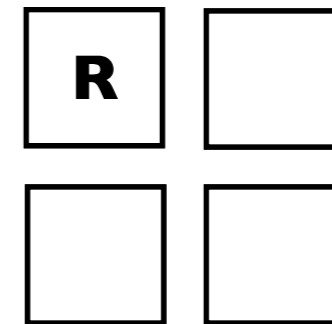
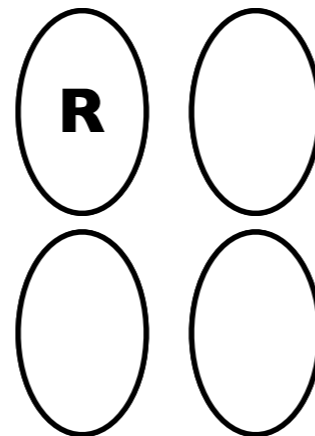
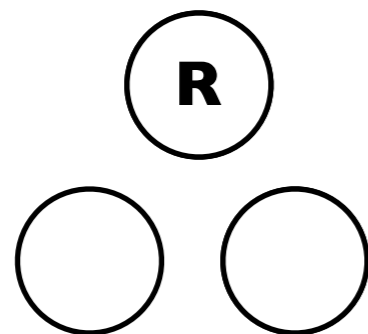
VENDOR



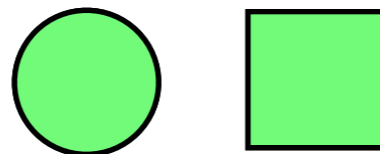
Staged deployment in Mirage

R: representatives

OUTSIDE
WORLD



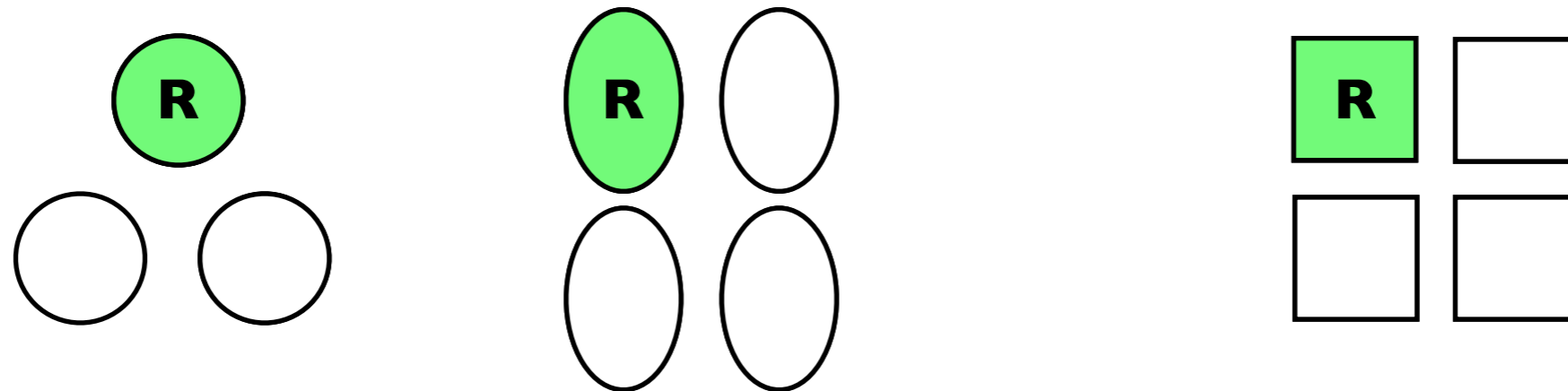
VENDOR



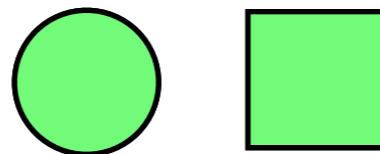
Staged deployment in Mirage

R: representatives

OUTSIDE
WORLD



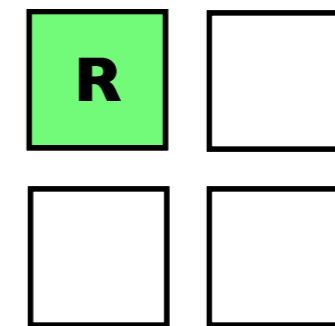
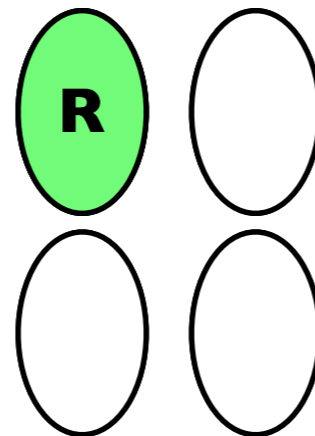
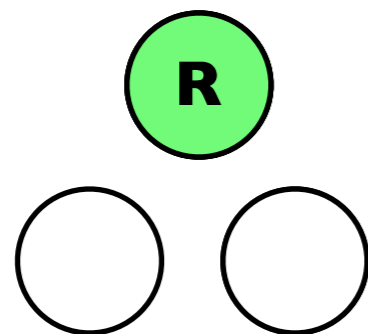
VENDOR



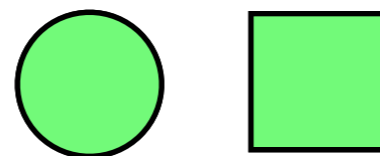
Staged deployment in Mirage

R: representatives

OUTSIDE
WORLD



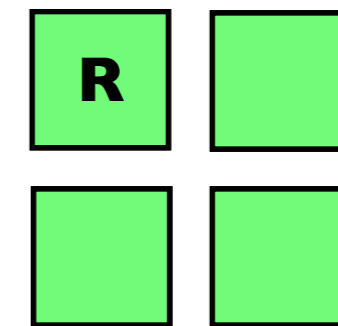
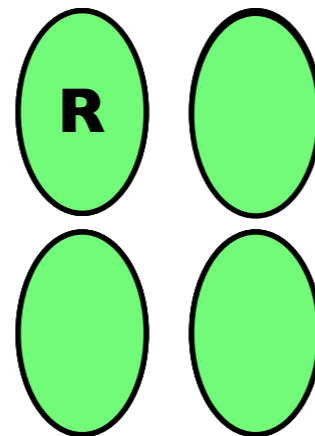
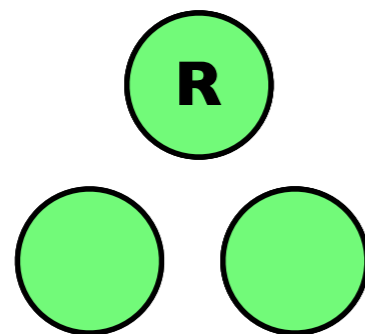
VENDOR



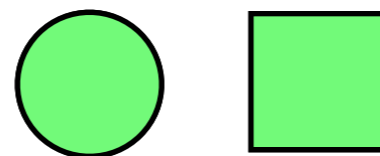
Staged deployment in Mirage

R: representatives

OUTSIDE
WORLD



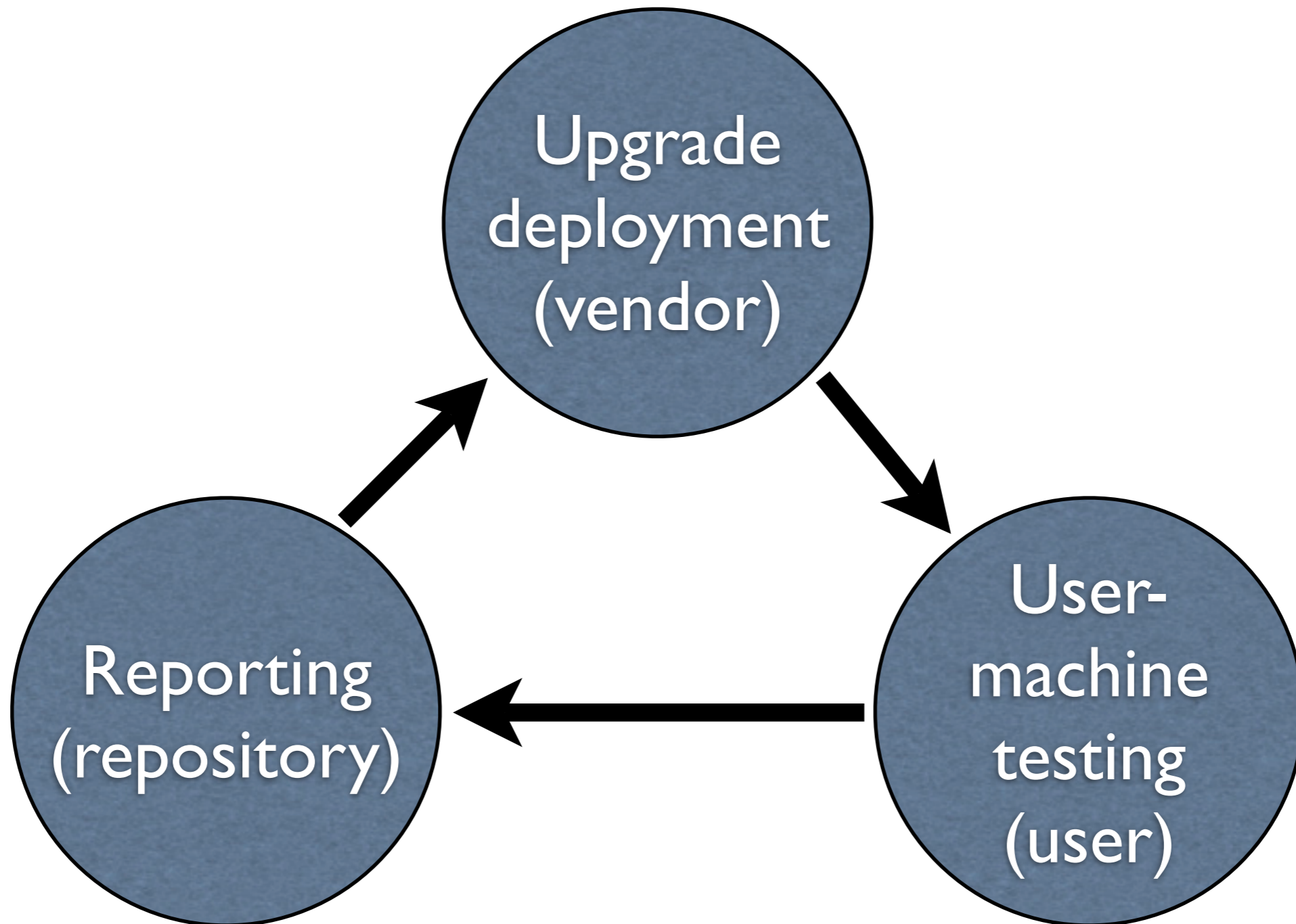
VENDOR



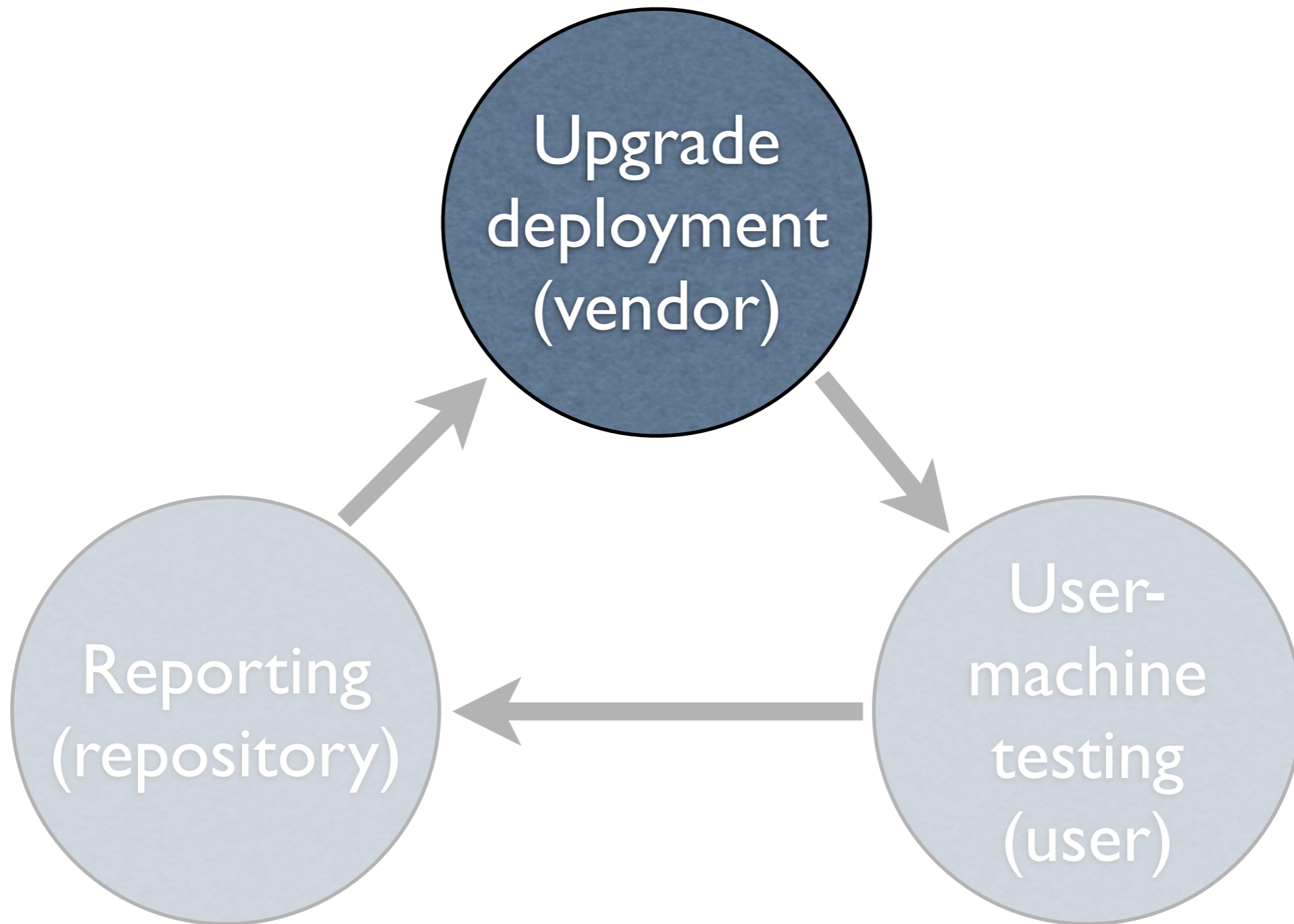
Key idea

- Leverage information about the environment:
 - ◆ better testing coverage
 - ◆ reduced upgrade overhead

Mirage



Mirage



Challenges

- Clustering machines
 - ✦ identify the environment
 - ✦ fingerprint environmental resources
 - ✦ cluster
- Deployment

Goal of clustering

- All machines in a cluster behave identically with respect to an upgrade

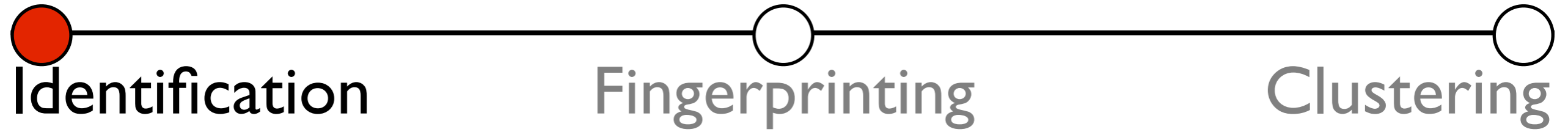
Benefit of clustering

- Testing at a representative provides information about an entire cluster

Extent of the benefit

- Depends on:
 - ✦ quality of clustering
 - ✦ quality of testing

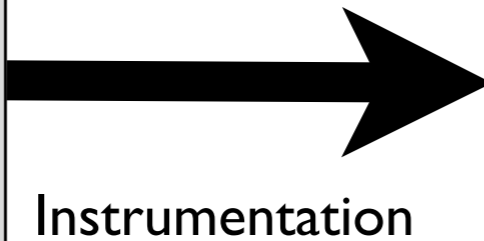
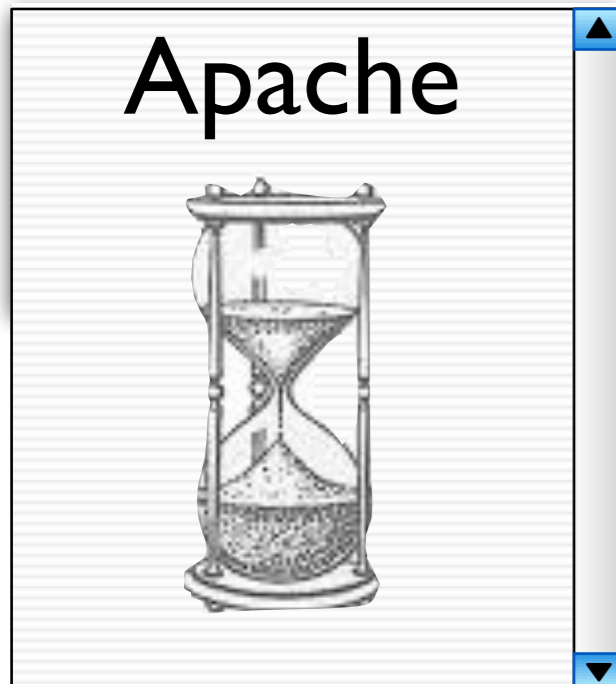
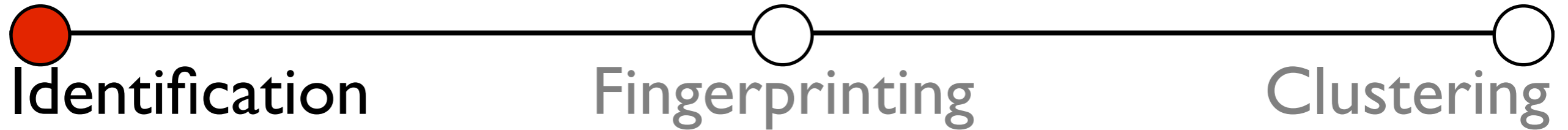
Clustering machines



Apache

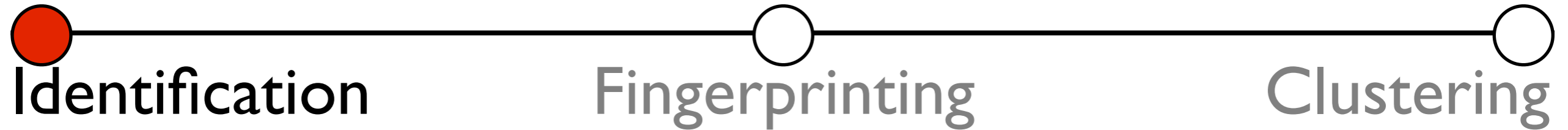


Clustering machines



```
libc-2.4.so  
$PATH  
httpd.conf  
access_log  
/tmp/xyz  
index.html  
Tomcat
```

Clustering machines



Libraries: `libc-2.4.so`

Environment variables: `$PATH`

Configuration files: `httpd.conf`

Log files: `access_log`

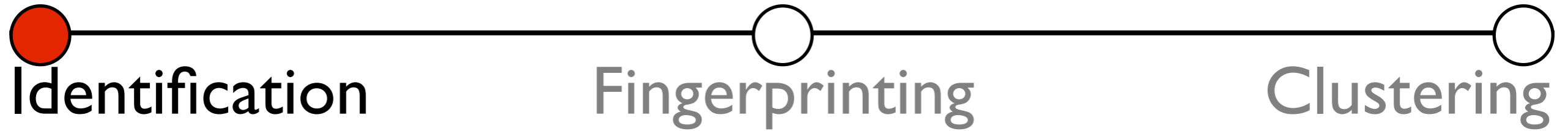
Temporary files: `/tmp/xyz`

Data files: `index.html`

Dependent Applications: `Tomcat`



Clustering machines



Libraries:

Environment variables:

Configuration files:

Log files:

Temporary files:

Data files:

Dependent Applications:

`libc-2.4.so`

`$PATH`

`httpd.conf`

`Tomcat`

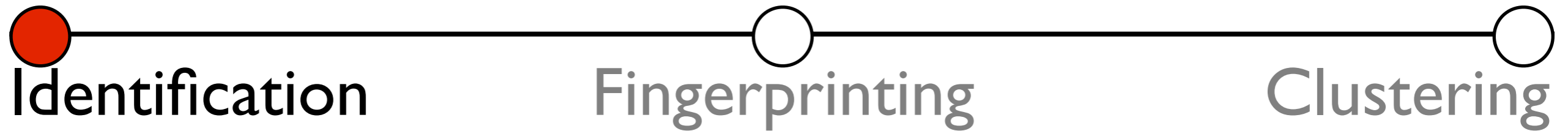
`access_log`

`/tmp/xyz`

`index.html`

heuristic +
vendor rules

Clustering machines



Libraries:

Environment variables:

Configuration files:

Log files:

Temporary files:

Data files:

Dependent Applications:

`access_log`

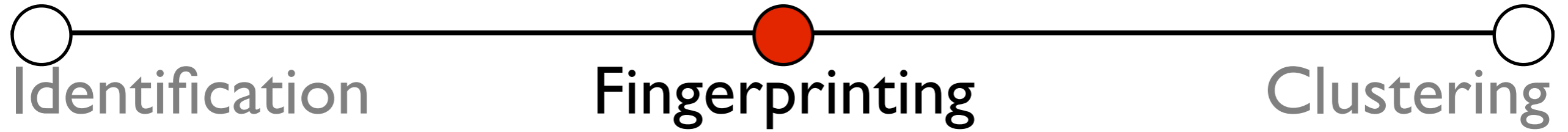
`/tmp/xyz`

`index.html`

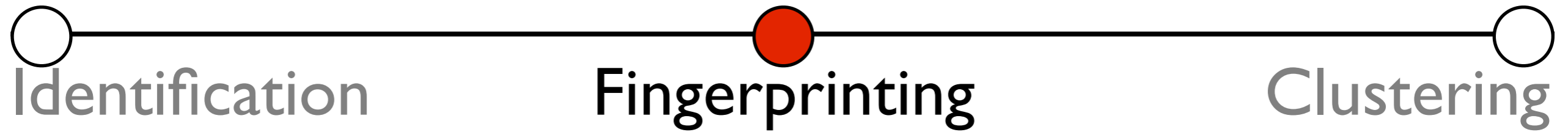


heuristic +
vendor rules

Clustering machines



Clustering machines



`libc-2.4.so`

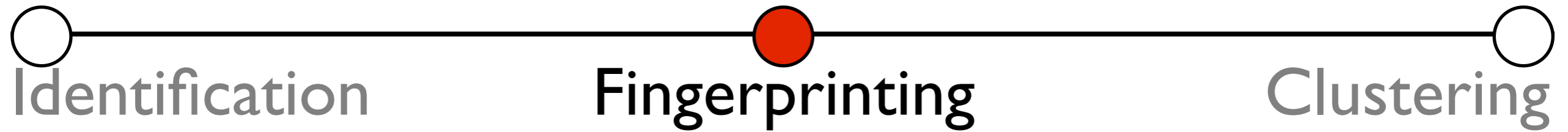
`$PATH`

`httpd.conf`

`Tomcat`

Parsers

Clustering machines

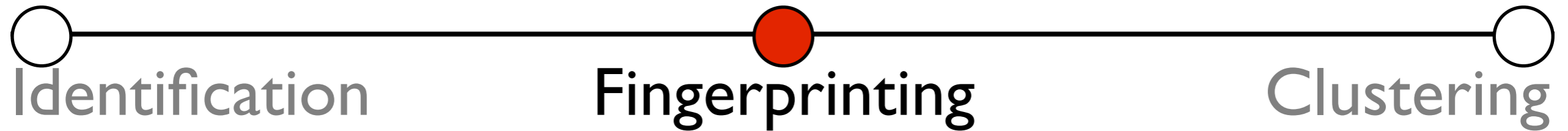


`$PATH`
`httpd.conf`
`Tomcat`

Parsers

`libc.2.4.hash`

Clustering machines



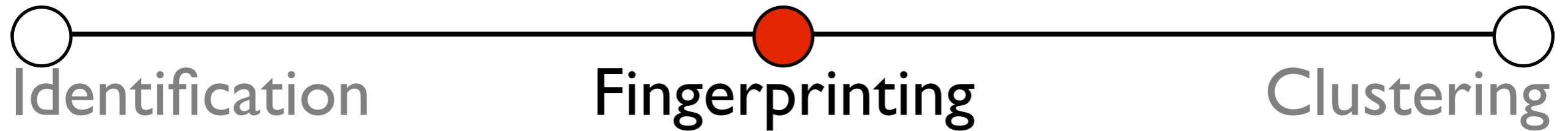
`httpd.conf`
`Tomcat`

Parsers

`libc.2.4.hash`

`$PATH.hash`

Clustering machines



Tomcat

Parsers

`libc.2.4.hash`

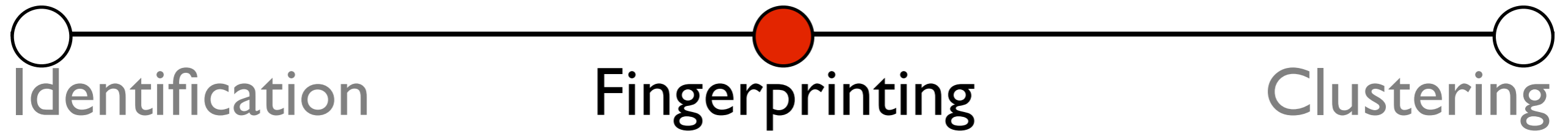
`$PATH.hash`

`httpd.conf.listen.hash`

`httpd.conf.mod_php.hash`

`httpd.conf.mod_cgi.hash`

Clustering machines



`libc.2.4.hash`

`$PATH.hash`

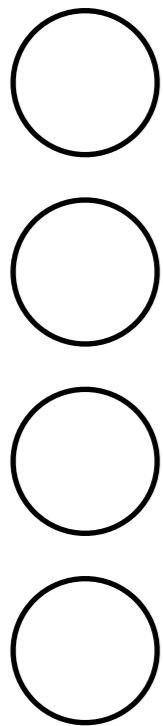
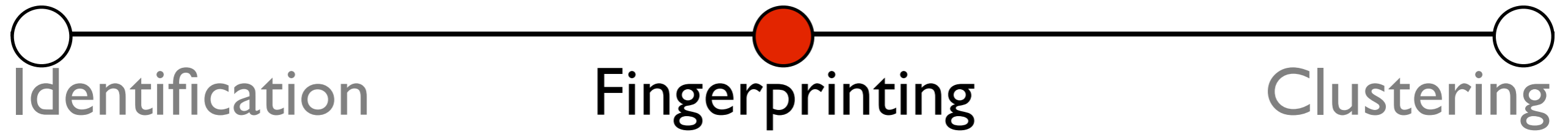
`httpd.conf.listen.hash`

`httpd.conf.mod_php.hash`

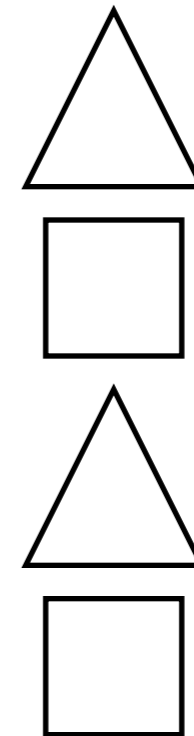
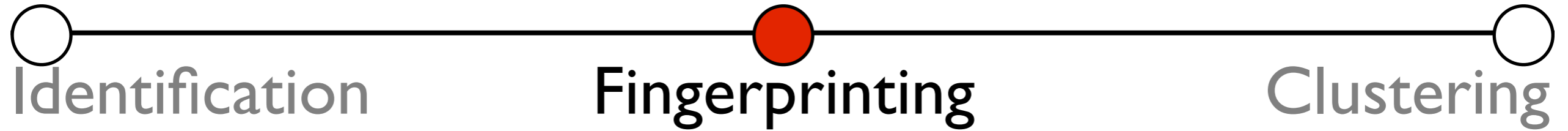
`httpd.conf.mod_cgi.hash`

`Tomcat.5.5`

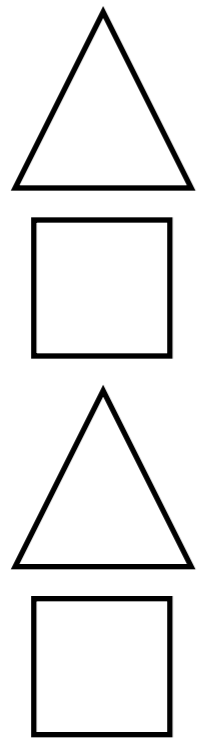
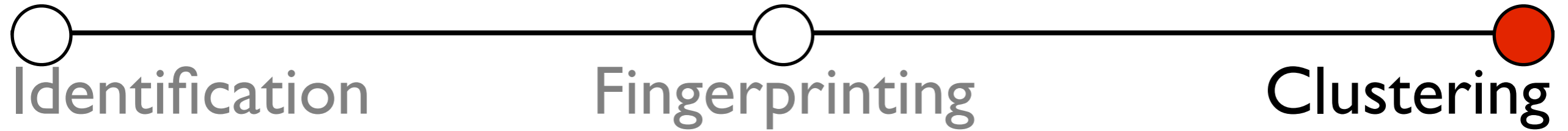
Clustering machines



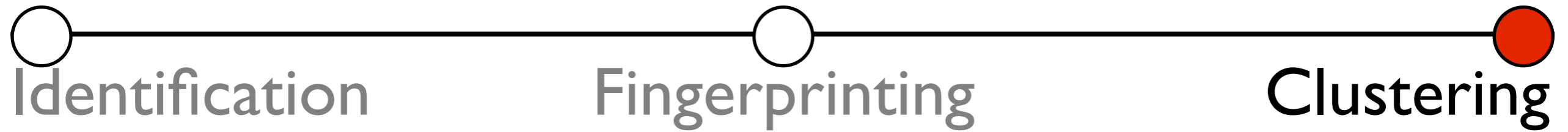
Clustering machines



Clustering machines



Clustering machines



Challenges

- Clustering machines
 - ◆ identify the environment
 - ◆ fingerprint environmental resources
 - ◆ cluster
- Deployment

Challenges

- Clustering machines
 - ◆ identify the environment
 - ◆ fingerprint environmental resources
 - ◆ cluster
- Deployment

Deployment goals

- Low upgrade overhead
- Fast deployment

Deployment goals

- Low upgrade overhead
- Fast deployment

There is a tradeoff !

Controlling the tradeoff

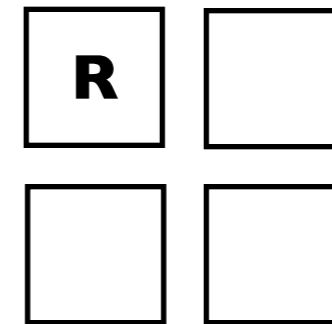
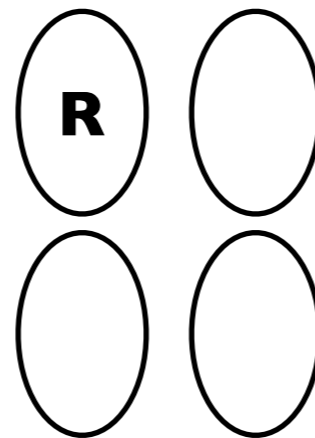
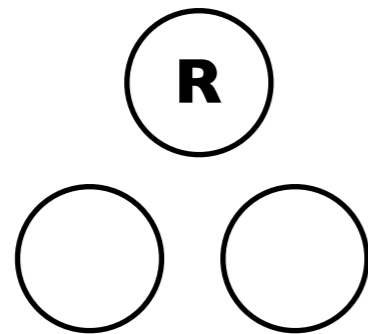
- Leverage clusters for staging
- Deploy in parallel or in sequential stages

	parallel	sequential
speed of deployment	✓	✗
upgrade overhead	✗	✓

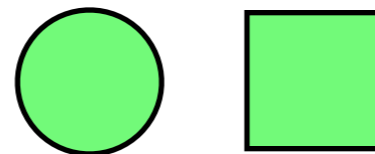
Representatives first, in parallel

R: representatives

OUTSIDE
WORLD



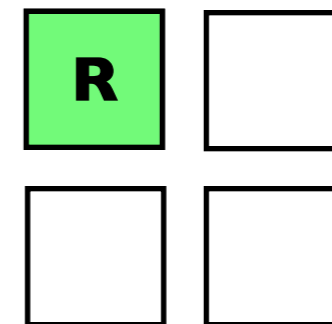
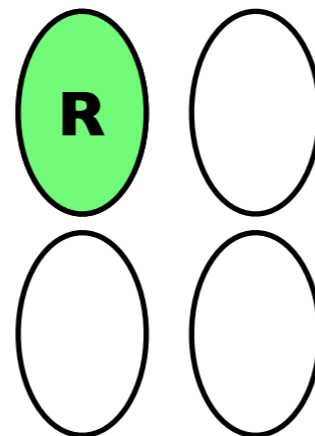
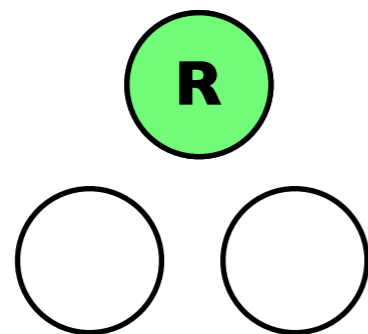
VENDOR



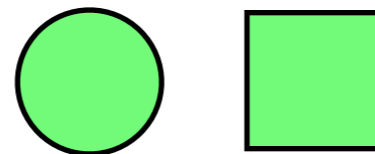
Representatives first, in parallel

R: representatives

OUTSIDE
WORLD



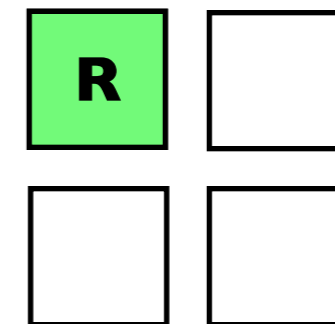
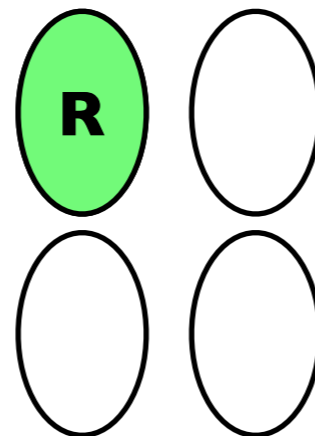
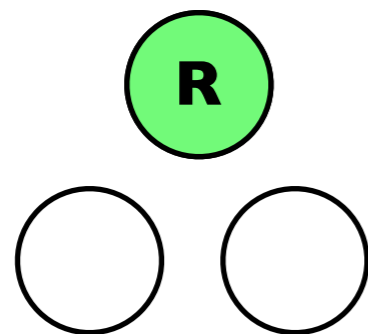
VENDOR



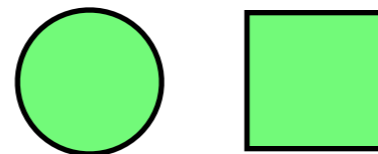
Representatives first, in parallel

R: representatives

OUTSIDE
WORLD



VENDOR

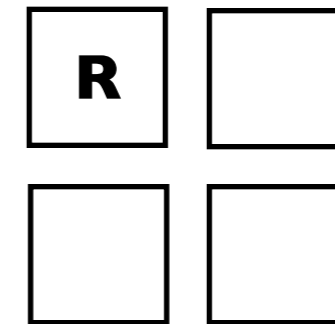
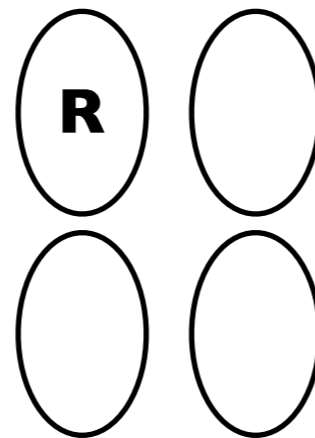
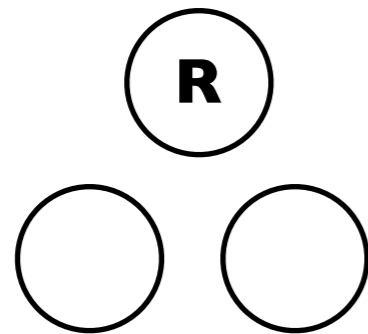


Upgrade overhead: 2

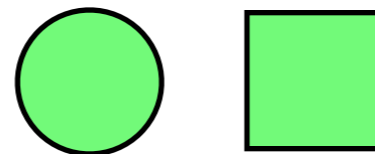
Representatives first, in sequence

R: representatives

OUTSIDE
WORLD



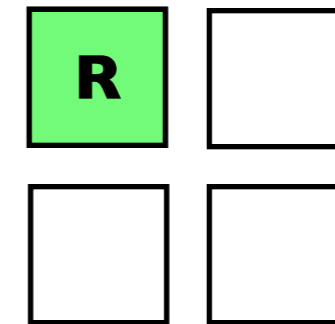
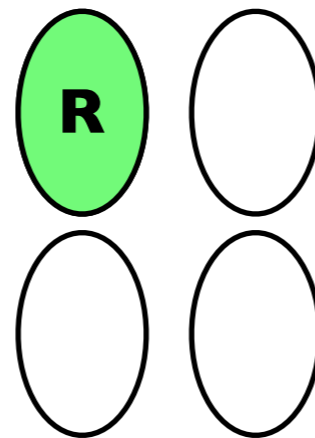
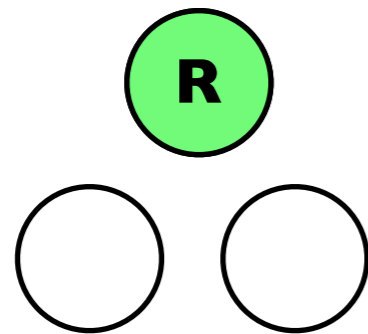
VENDOR



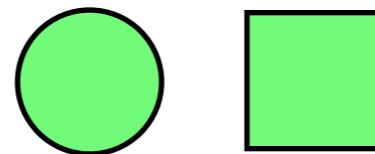
Representatives first, in sequence

R: representatives

OUTSIDE
WORLD



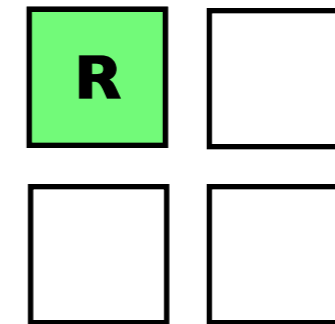
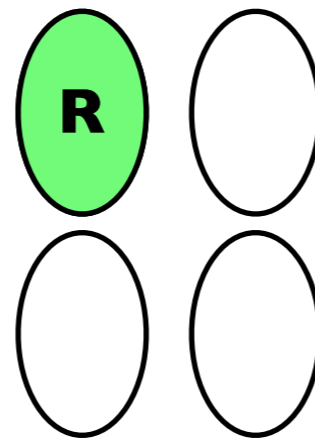
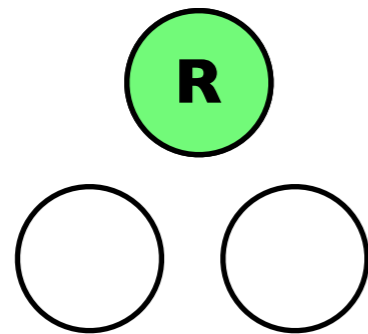
VENDOR



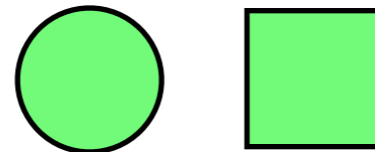
Representatives first, in sequence

R: representatives

OUTSIDE
WORLD



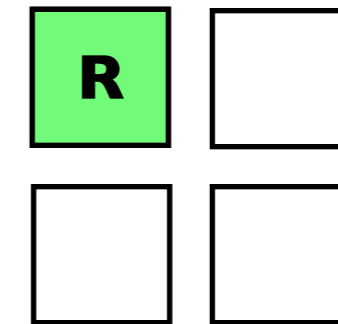
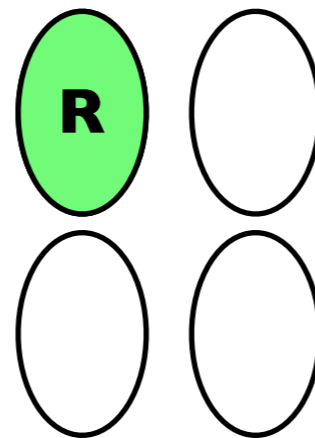
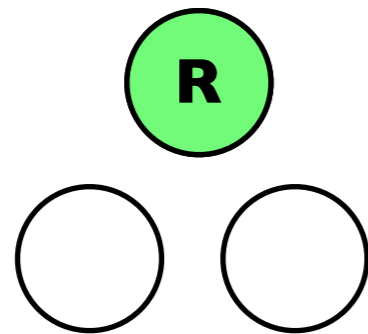
VENDOR



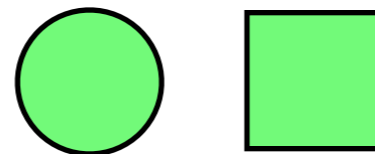
Representatives first, in sequence

R: representatives

OUTSIDE
WORLD



VENDOR

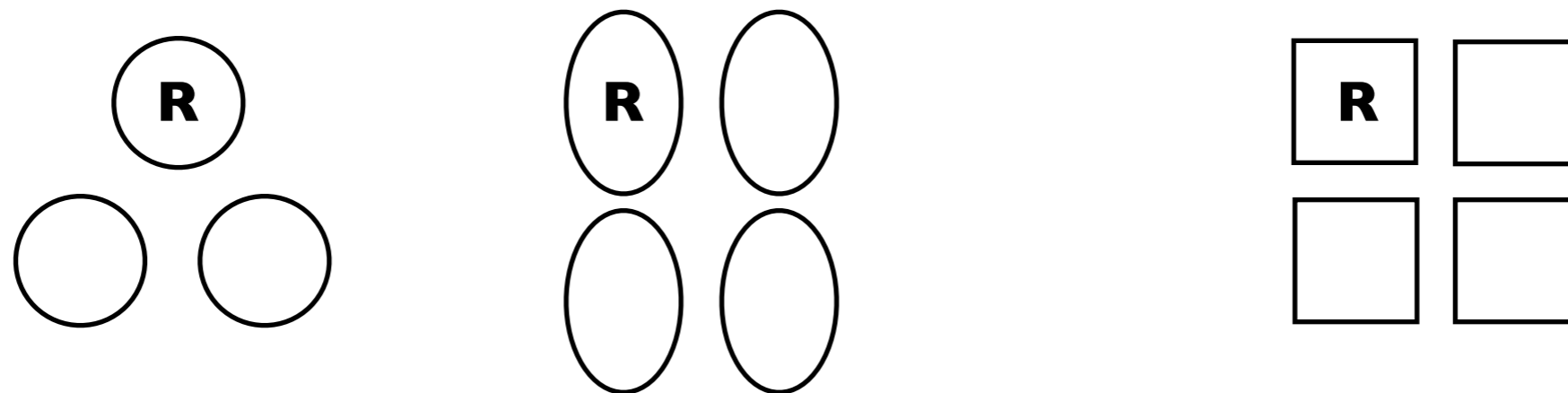


Upgrade overhead: 1

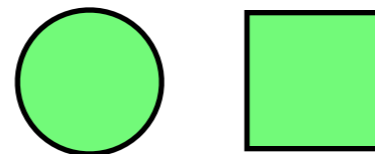
Representatives and other machines in parallel

R: representatives

OUTSIDE
WORLD



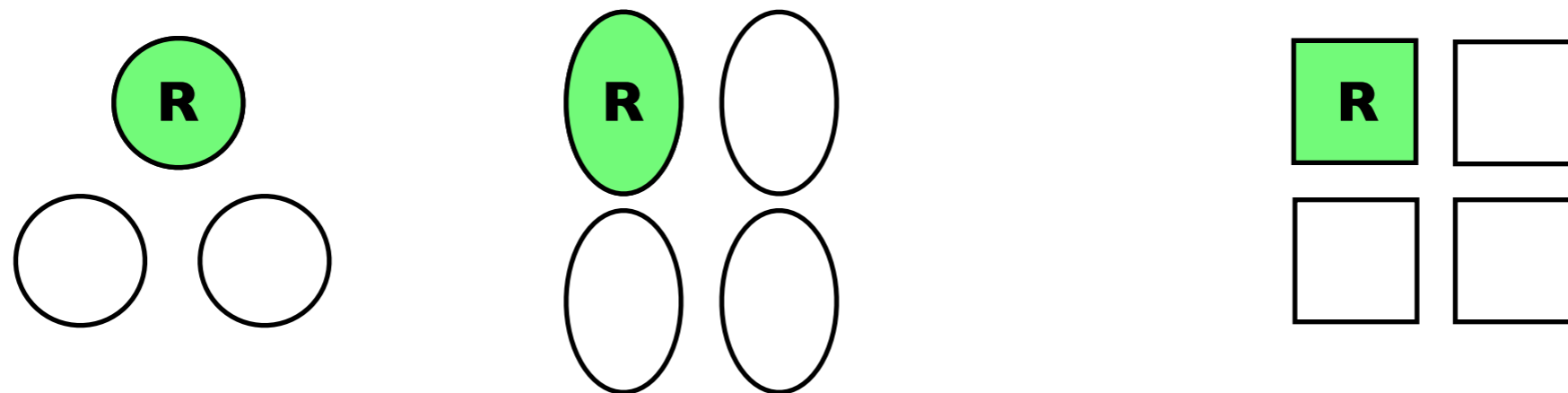
VENDOR



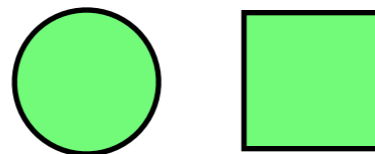
Representatives and other machines in parallel

R: representatives

OUTSIDE
WORLD



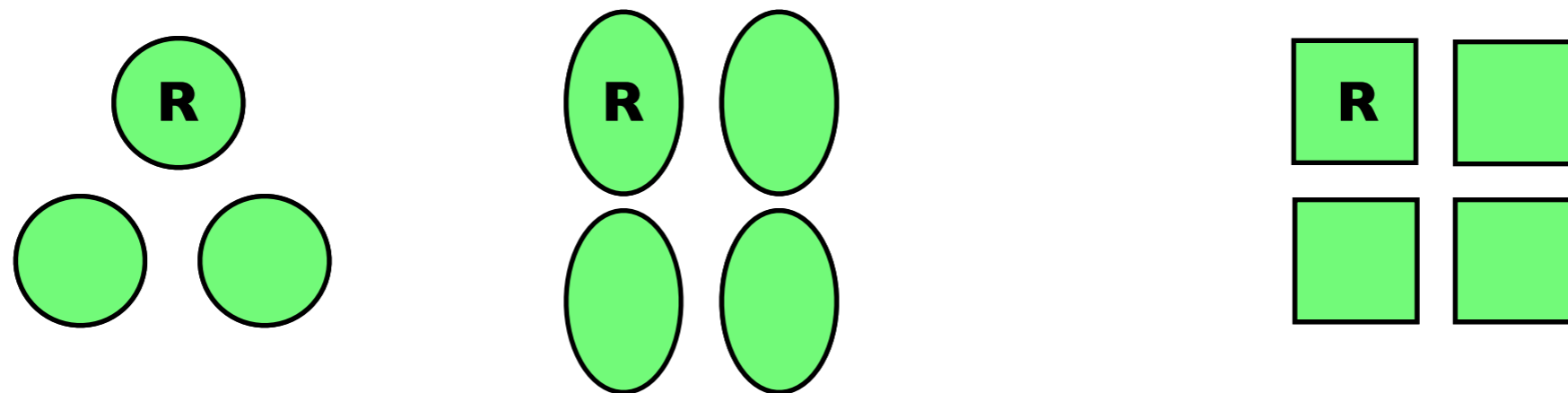
VENDOR



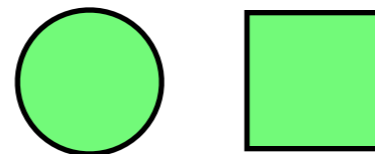
Representatives and other machines in parallel

R: representatives

OUTSIDE
WORLD



VENDOR

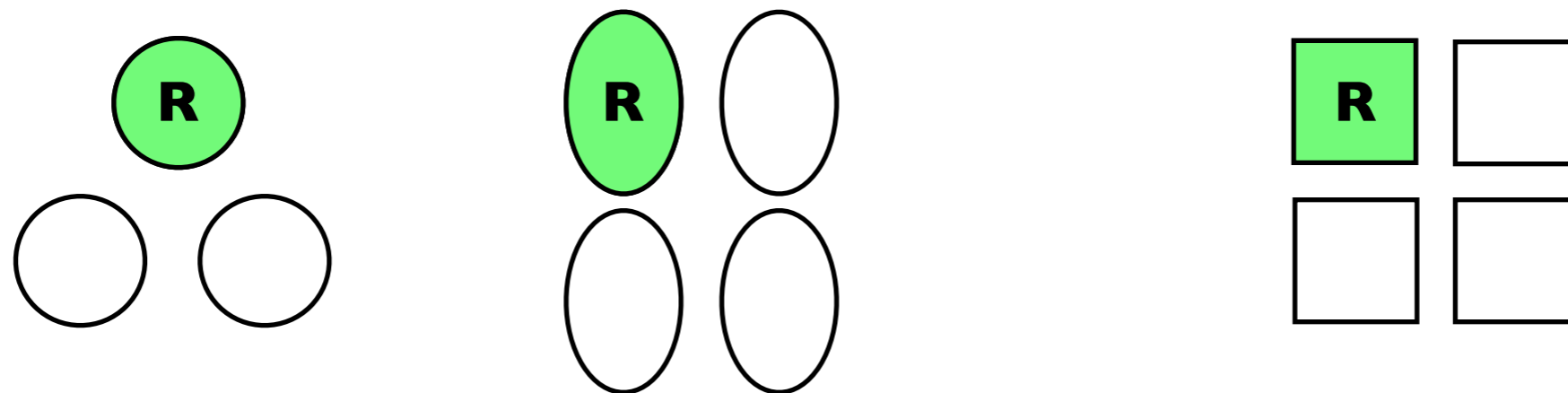


Upgrade overhead: 2

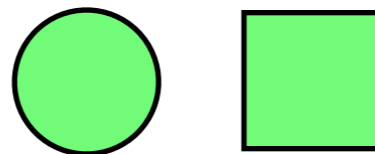
Representatives and other machines in parallel

R: representatives

OUTSIDE
WORLD



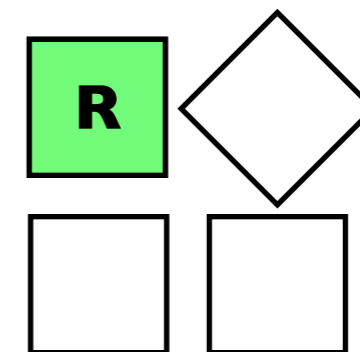
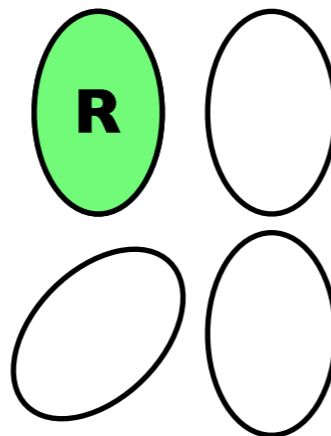
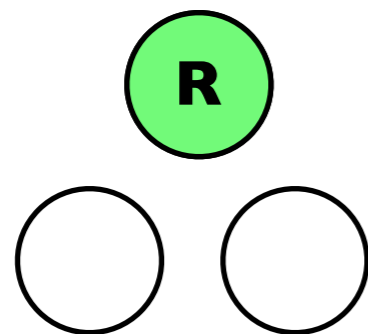
VENDOR



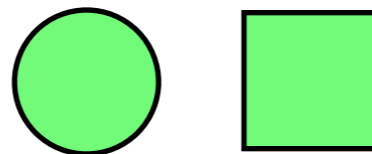
Representatives and other machines in parallel

R: representatives

OUTSIDE
WORLD



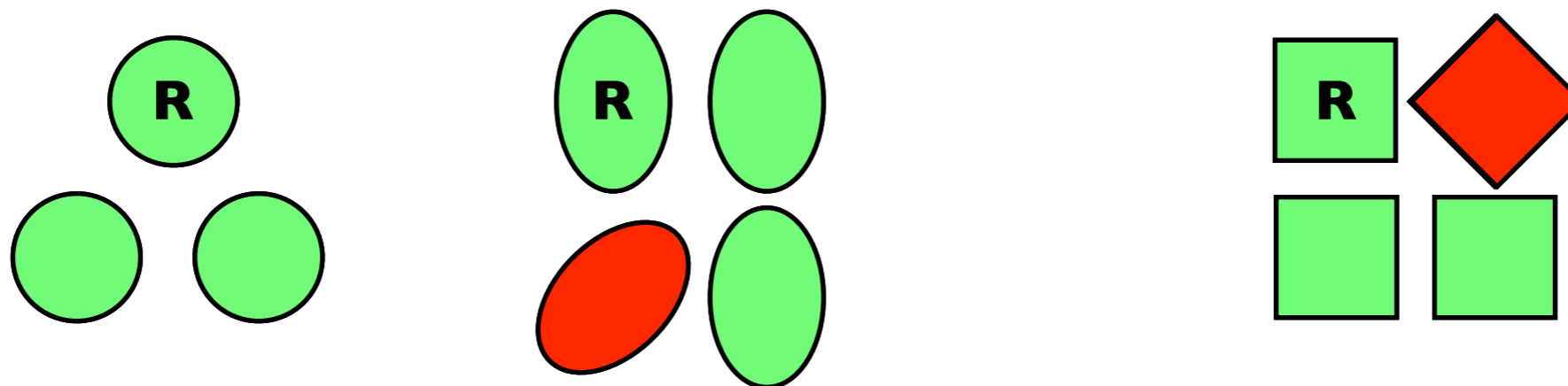
VENDOR



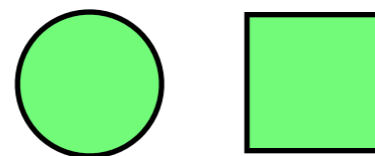
Representatives and other machines in parallel

R: representatives

OUTSIDE
WORLD



VENDOR

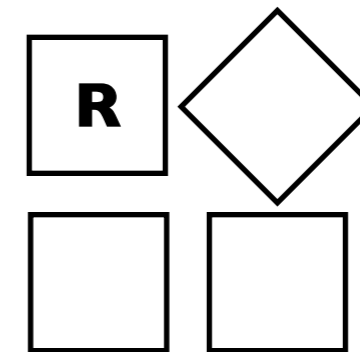
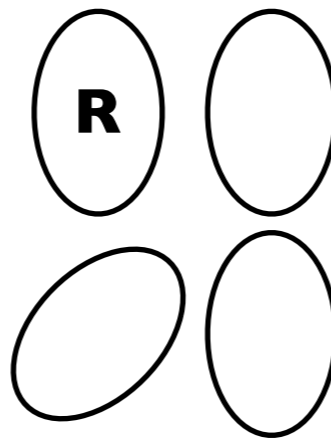
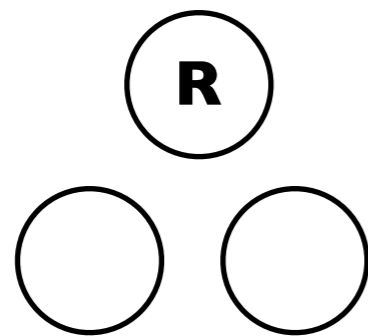


Upgrade overhead: 4

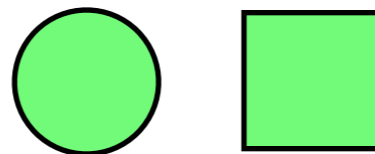
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



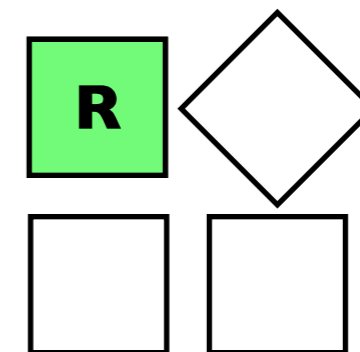
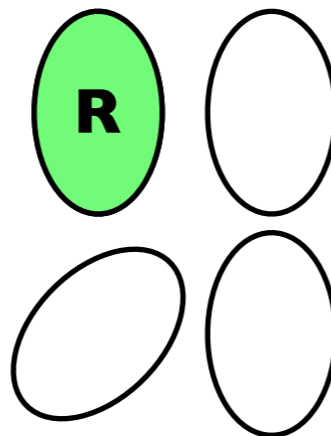
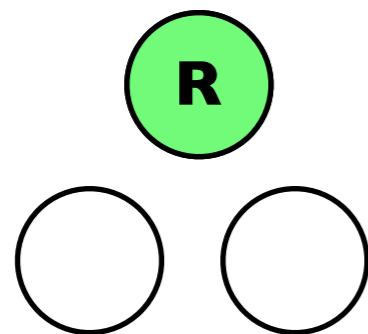
VENDOR



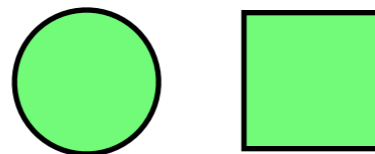
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



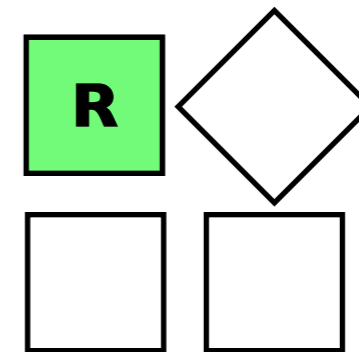
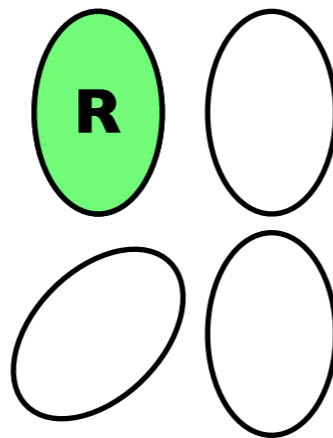
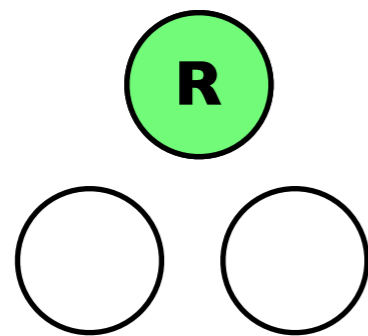
VENDOR



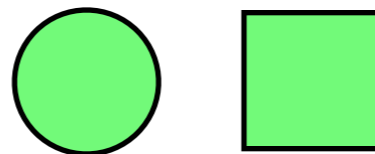
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



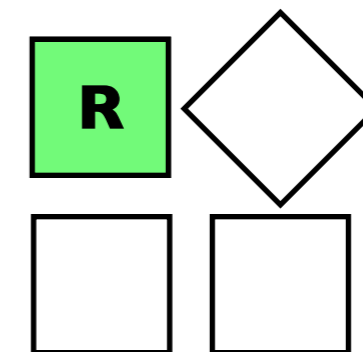
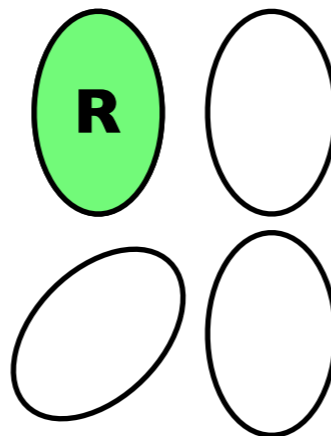
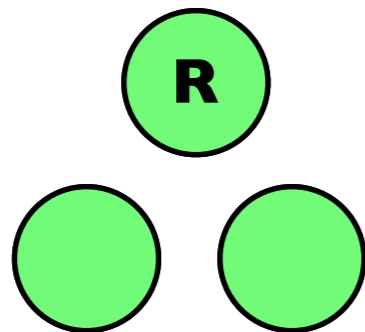
VENDOR



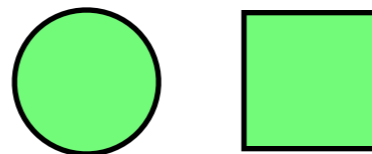
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



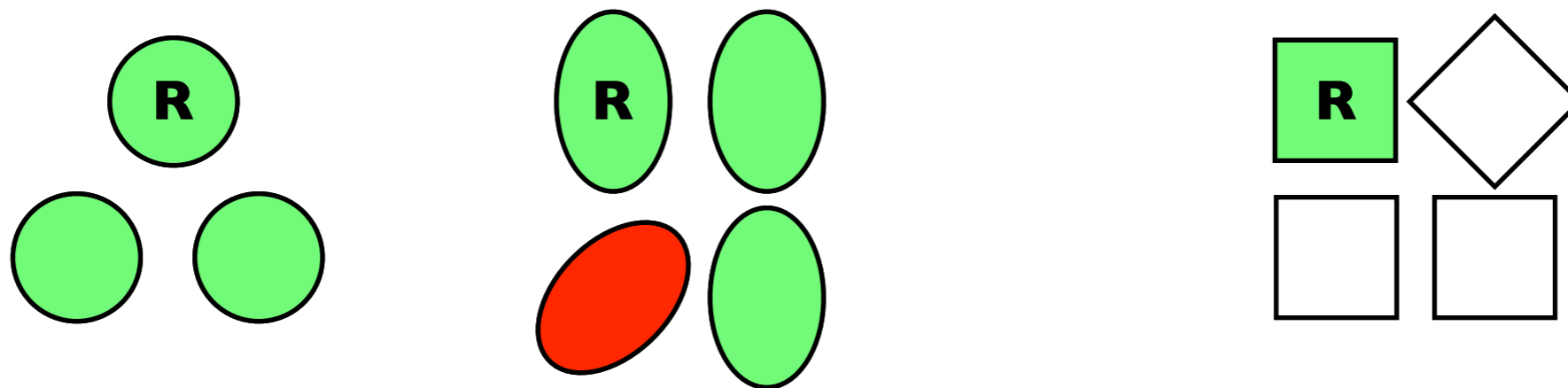
VENDOR



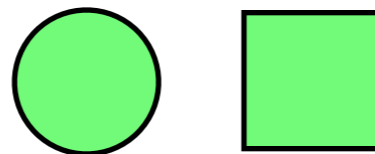
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



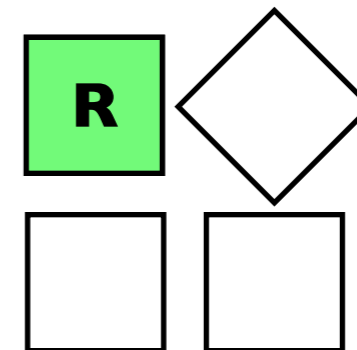
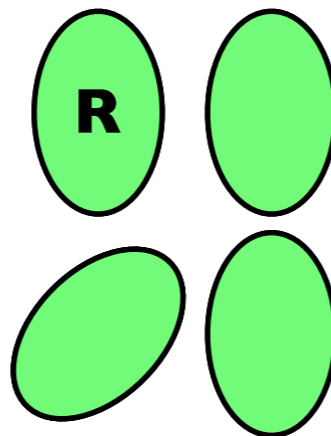
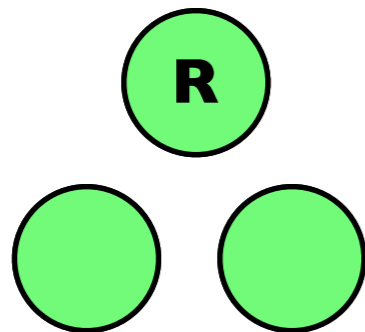
VENDOR



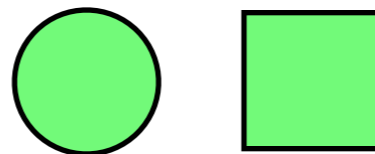
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



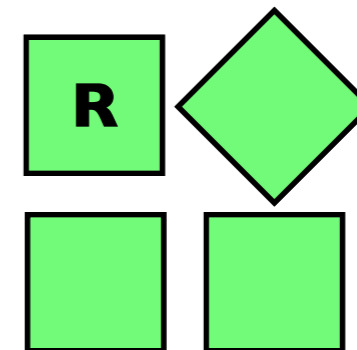
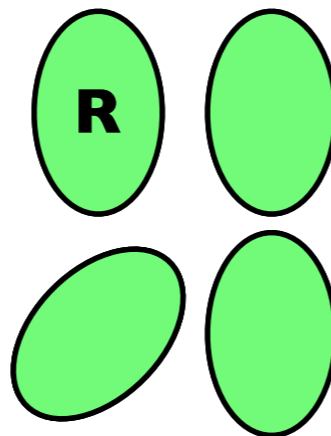
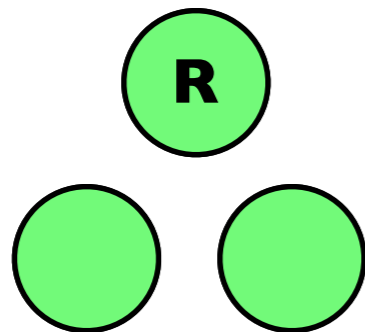
VENDOR



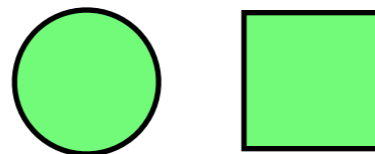
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



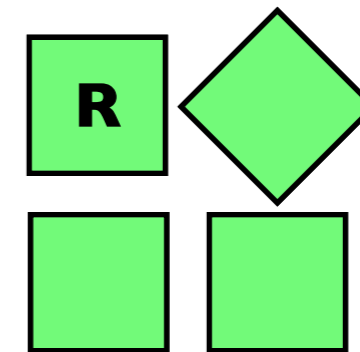
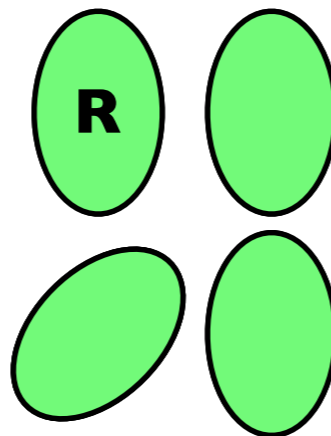
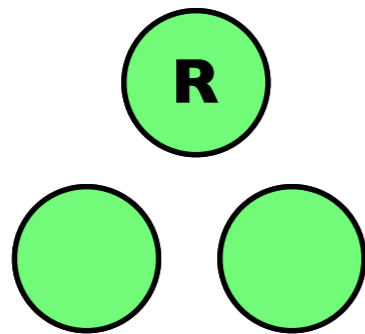
VENDOR



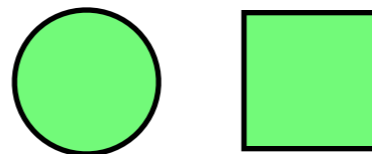
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



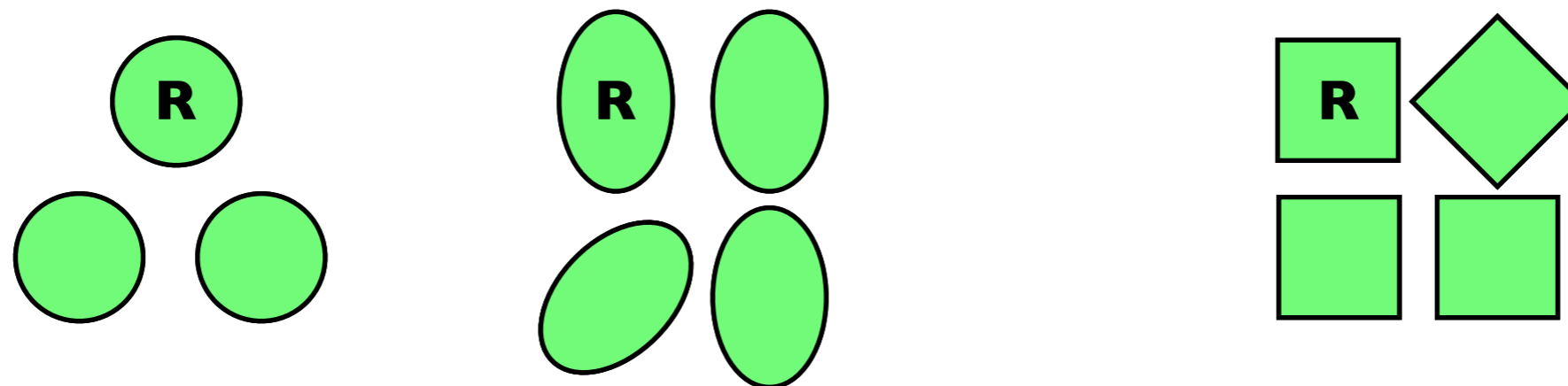
VENDOR



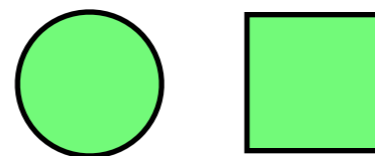
Representatives in parallel, other machines in sequence

R: representatives

OUTSIDE
WORLD



VENDOR



Upgrade overhead: 3

Evaluation

- Quality of clustering:
 - ✦ accuracy of identification of env. resources
 - ✦ accuracy of the clustering algorithm
- Staged deployment:
 - ✦ control upgrade overhead and deployment speed

Can we accurately identify environmental resources ?

App.	# env. resources	# vendor rules	#errors
firefox	839	7	0
apache	251	2	0
php	206	0	0
mysql	250	1	0

Can we accurately identify environmental resources ?

App.	# env. resources	# vendor rules	#errors
firefox	839	7	0
apache	251	2	0
php	206	0	0
mysql	250	1	0

Example: Apache

- Files initially misclassified:
 - ✦ `index.html`
 - ✦ `access_log`
- Vendor rules needed:
 - ✦ `exclude *.html`
 - ✦ `exclude access_log`

Can we accurately cluster machines?

- 21 MySQL environments
 - ◆ 2 distributions of linux
 - ◆ PHP and Apache
 - ◆ various MySQL configurations
- 2 real upgrade problems

Can we accurately cluster machines?

- 0 misplaced machines
 - ✦ optimal for reducing upgrade overhead by testing at representatives
- 15 clusters
 - ✦ artefact of the experiment
 - ✦ number of clusters can be reduced by increasing the fingerprinting granularity

Controlling the tradeoff

- Experimental setup:
 - ◆ Event-driven simulator
 - ◆ 100'000 machines
 - ◆ 3 problems
 - ◆ 2 staging protocols, plus NoStaging

Controlling the tradeoff

- Upgrade overhead reduced very significantly
- Deployment completes 25% later than NoStaging in the worst case

Related work

- Package management systems
- Patch management systems
- Strider, Microsoft ACT
- No other work considered clustering and staged deployment

Conclusion

- Staged deployment in Mirage
 - ✦ machines can be clustered based on their environment
 - ✦ significantly reduce upgrade overhead

Thank you for your attention !

olivier.crameri@epfl.ch

Travel scholarship provided by: 