DEPLOYMENT AND RECONFIGURATION CHALLENGES

Dimitri Pertin, Hélène Coullon, Christian Perez

Ascola & Avalon

IPL Discovery

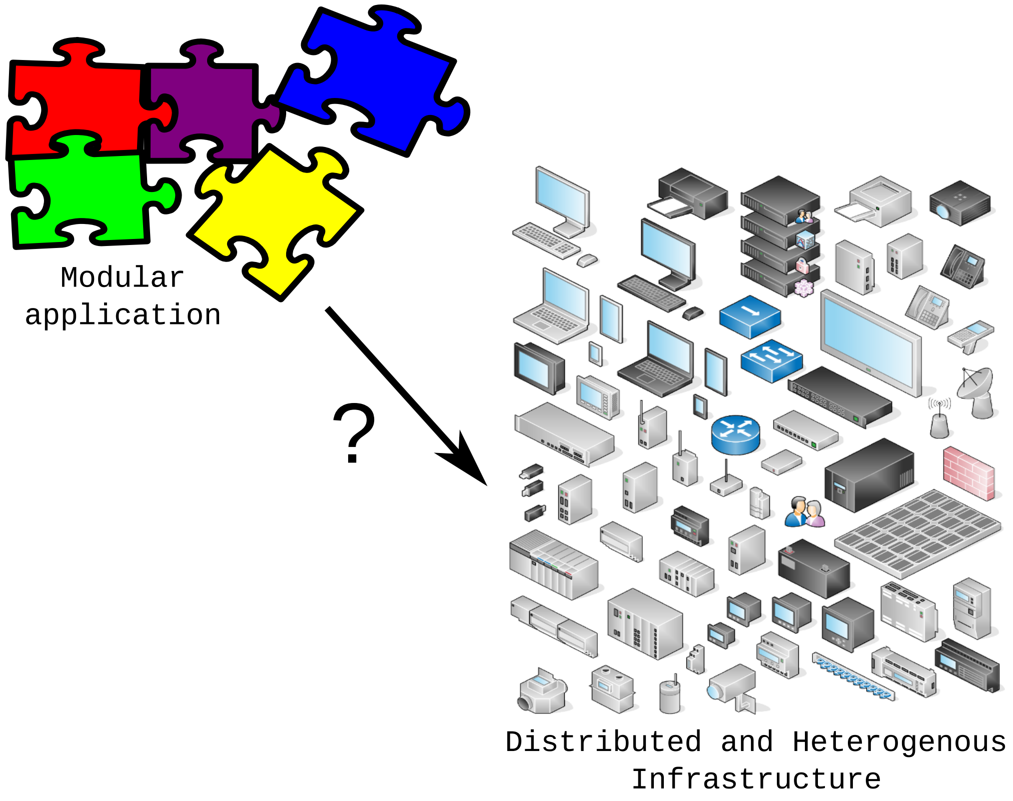
beyondtheclouds.github.io

OUTLINE

MOTIVATION

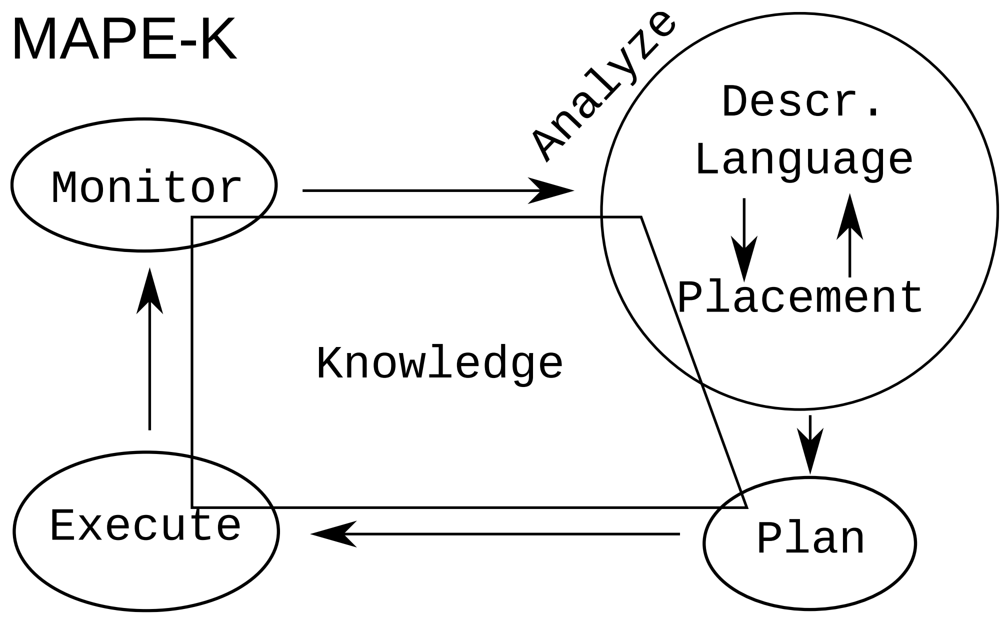
CONTEXT

DEPLOYMENT AUTOMATION



DEPLOYMENT / RECONFIGURATION

MAPE-K

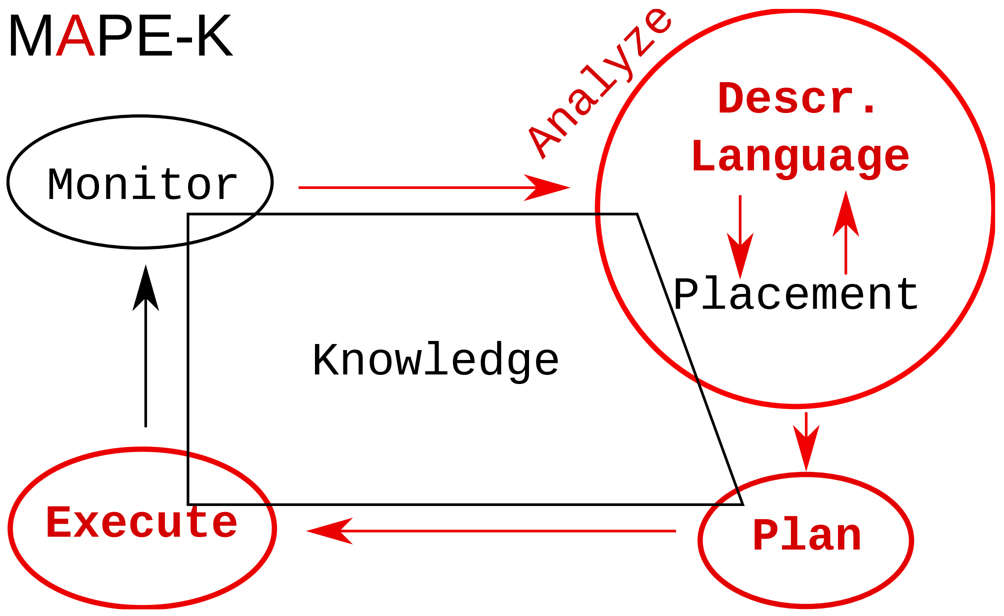


ANALYZE

- Descr. Language
  - *What* to deploy?
  - *How* to deploy?
- Placement
  - *Where* to deploy?
  - Infrastructure/resource description

DEPLOYMENT / RECONFIGURATION

FOCUS



BIG PICTURE

How to deploy/re-deploy systems and applications on infrastructures?

Expected properties:

- low-level flexible generic model
- appropriate level of expressivity
- reliability and correctness
- reconfiguration
- performances and scalability

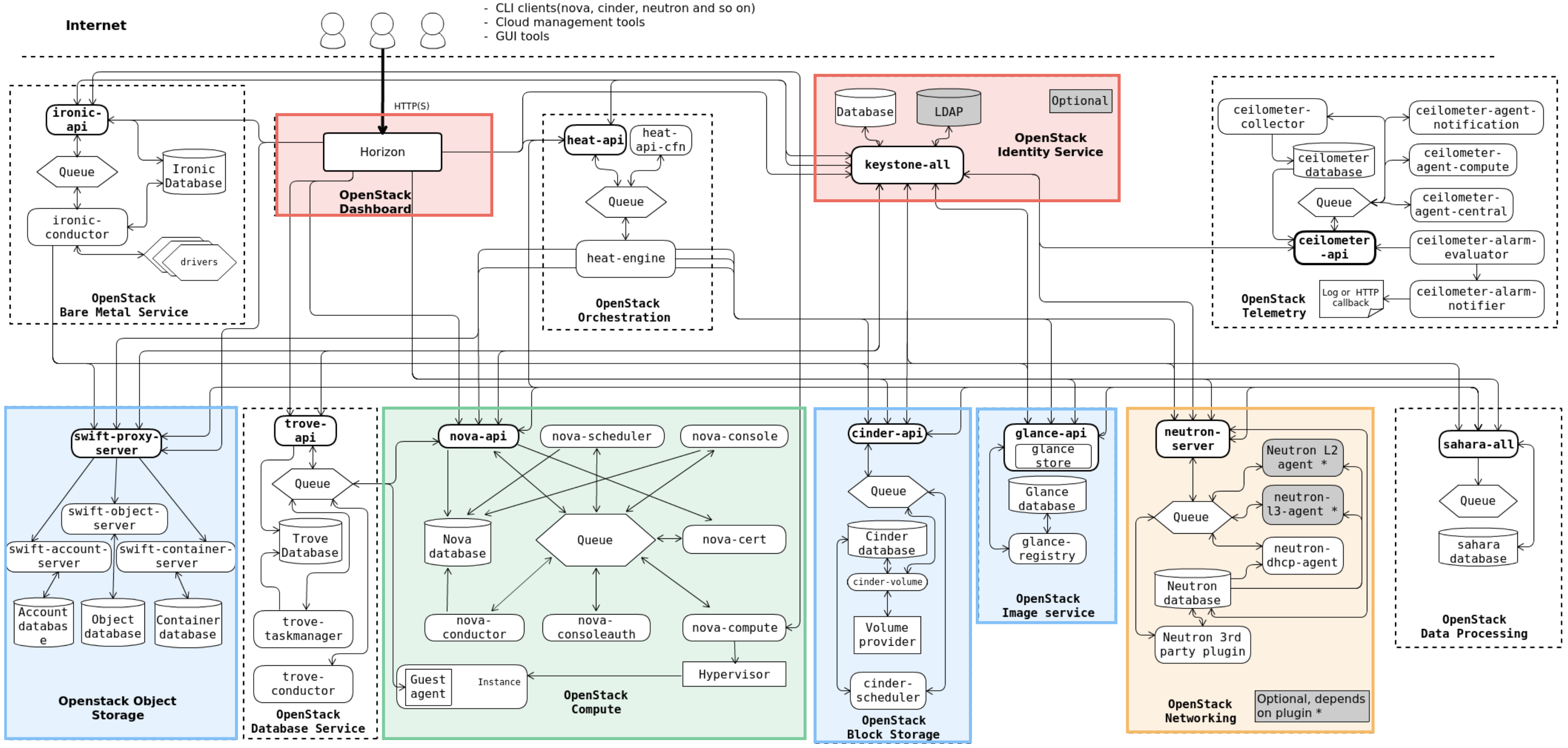
TWO PHASES

1. Research on the initial deployment problem
   - Dimitri Pertin, postdoc researcher
2. Research on the reconfiguration problem
   - Maverick Chardet, PhD student (October 2017)

USE CASES

Any application or system

- OpenStack and its decentralized version



USE CASES

- Smart-* applications composed of hybrid services
  - BigData
  - HPC
  - Stream processing
  - Virtual reality
  - etc.

SURVEY

SURVEY

Hélène Coullon, Christian Perez, Dimitri Pertin. *Production Deployment Tools for IaaSes: an Overall Model and Survey* - FiCloud 2017

PRODUCTION TOOLS

1. Kolla: *deploy production-ready OpenStack instances by leveraging Ansible and Docker*
2. Juju: *Canonical project to write your application life cycle and deploy it on major cloud providers*
3. Kubernetes: *a project designed by Google to deploy and maintain containerized applications*
4. TripleO (OpenStack On OpenStack): *an OpenStack project aiming at deploying OpenStack instances using OpenStack's own services*

PRODUCTION TOOLS

GENERICITY

	Kolla	Juju	K8s	TripleO
app. generic	No	Yes	Yes	No
env. generic	No	Yes	No	No

PRODUCTION TOOLS

PLACEMENT ET MONITOR

	Kolla	Juju	K8s	TripleO
Plac.	Ext.	Ext.	Int.	Int.
	Manual	Auto	Auto	Auto
Mntr.	No	Manual	1/2 Auto	1/2 Auto

PRODUCTION TOOLS

DESCRIPTION LANGUAGE

	Kolla	Juju	K8s	TripleO
expressivity	3.	1.	1.	2.
reliability	Retry	Retry	Retry	Retry
reconfiguration	No	3.	1.	2.
performances	?	?	?	?

PRODUCTION TOOLS

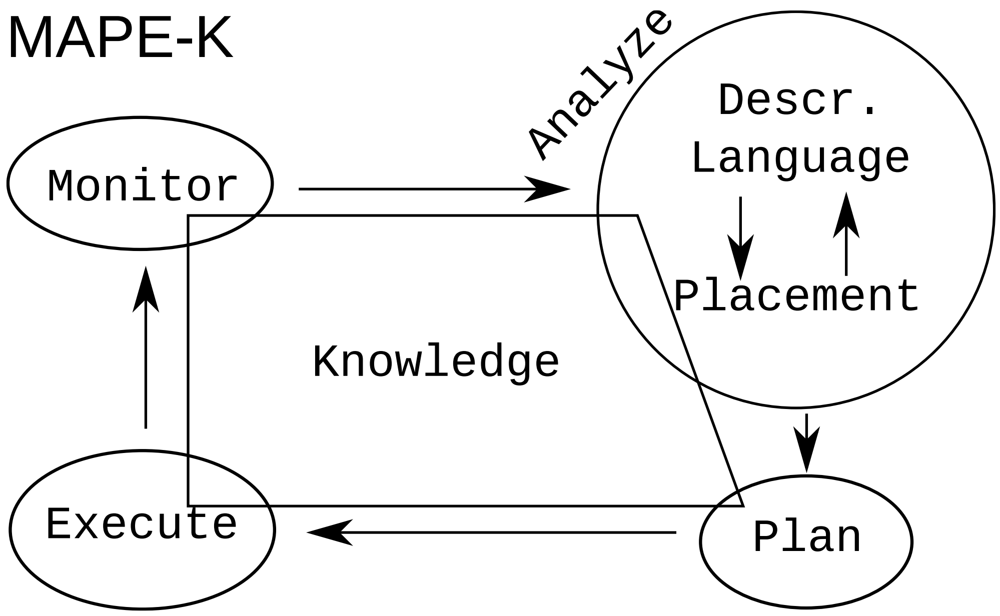
CONCLUSION

Expected properties:

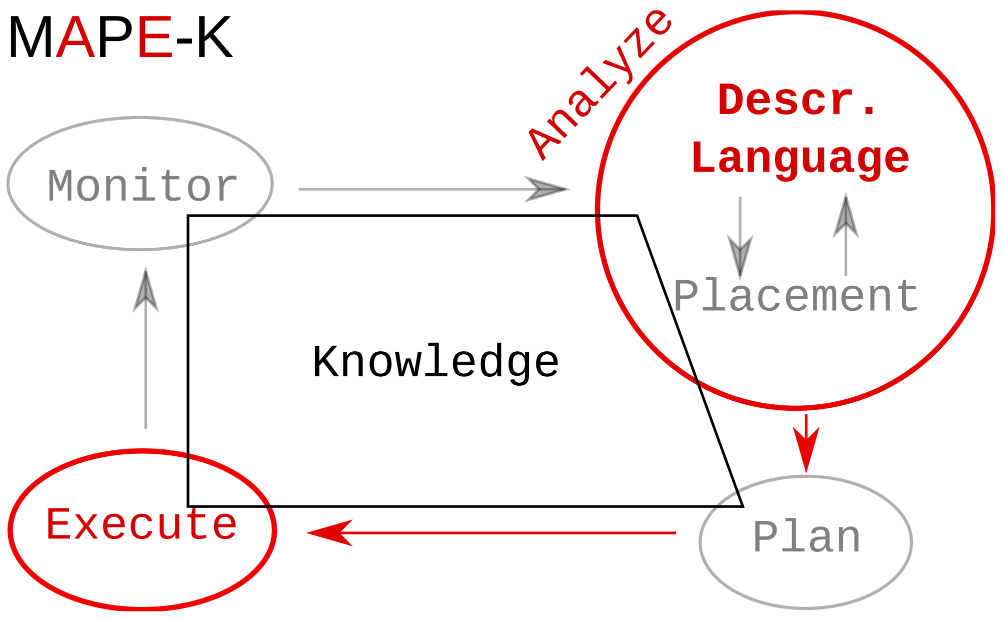
- low-level flexible generic model
- appropriate level of expressivity
- reliability and correctness
- reconfiguration
- performances and scalability

MAD MODEL

INITIAL DEPLOYMENT



INITIAL DEPLOYMENT

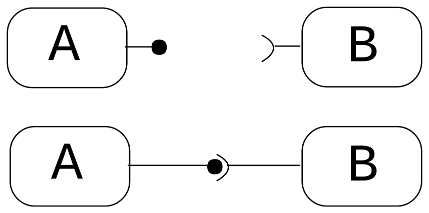


RELATED WORK

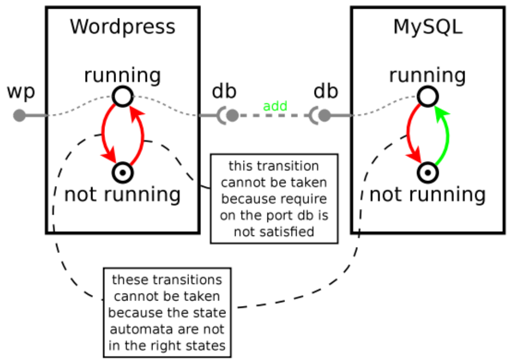
- Production tools: Juju, K8s, ansible
- Academic:
  - Tosca: ADL standard without associated semantic
  - Cloudify / OpenTosca:
    - Tosca engines
    - Answer *what?* and *where?*
  - Deployware: based on Fractal Component Model
  - Aeolus: Component Model + State-machine

COMPONENT MODELS

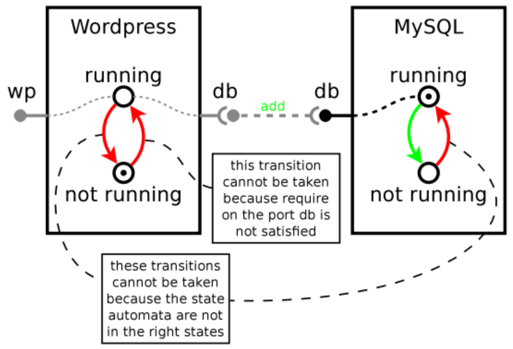
- *component* = module (type to instantiate)
- *use-provide ports* = functional dependencies between components
- *properties*: code-reuse, sep. concerns, reconfiguration



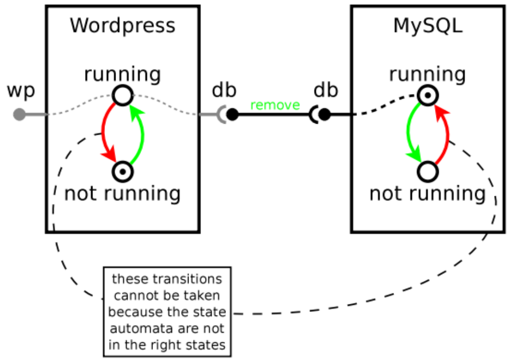
Di Cosmo et al. 2014, *Information and Computation*


Wordpress
running
not running
wp
db
add
db
MySQL
running
not running
this transition cannot be taken because require on the port db is not satisfied
these transitions cannot be taken because the state automata are not in the right states

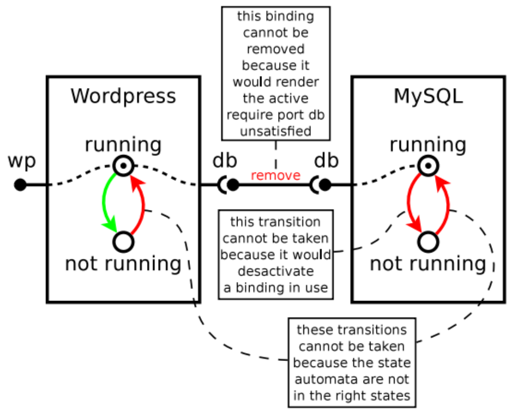
(a) Step 1: both components are not running, there is no binding between them.


Wordpress
running
not running
wp
db
add
db
MySQL
running
not running
this transition cannot be taken because require on the port db is not satisfied
these transitions cannot be taken because the state automata are not in the right states

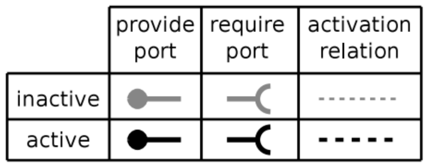
(b) Step 2: Only MySQL is running, there is still no binding.


Wordpress
running
not running
wp
db
remove
db
MySQL
running
not running
these transitions cannot be taken because the state automata are not in the right states

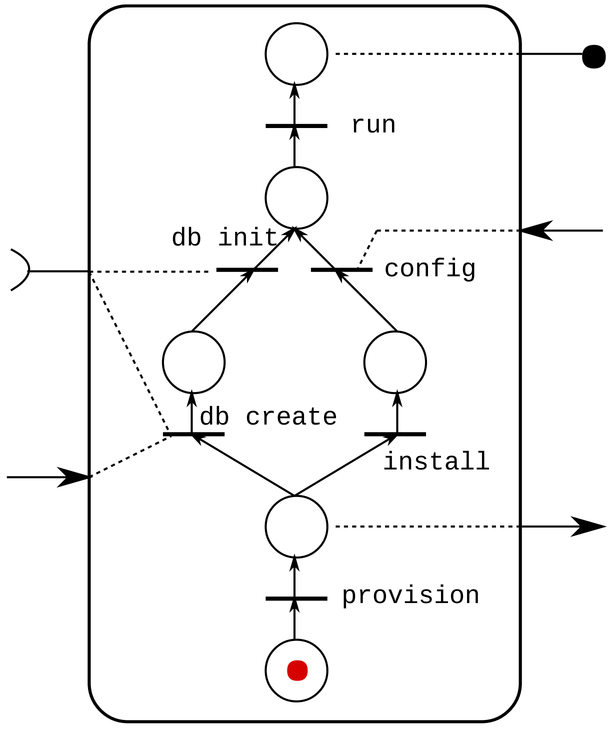
(c) Step 3: Only MySQL is running, a binding has been established.


this binding cannot be removed because it would render the active require port db unsatisfied
Wordpress
running
not running
wp
db
remove
db
MySQL
running
not running
this transition cannot be taken because it would desactivate a binding in use
these transitions cannot be taken because the state automata are not in the right states

(d) Step 4: Both components are running, the binding is present.

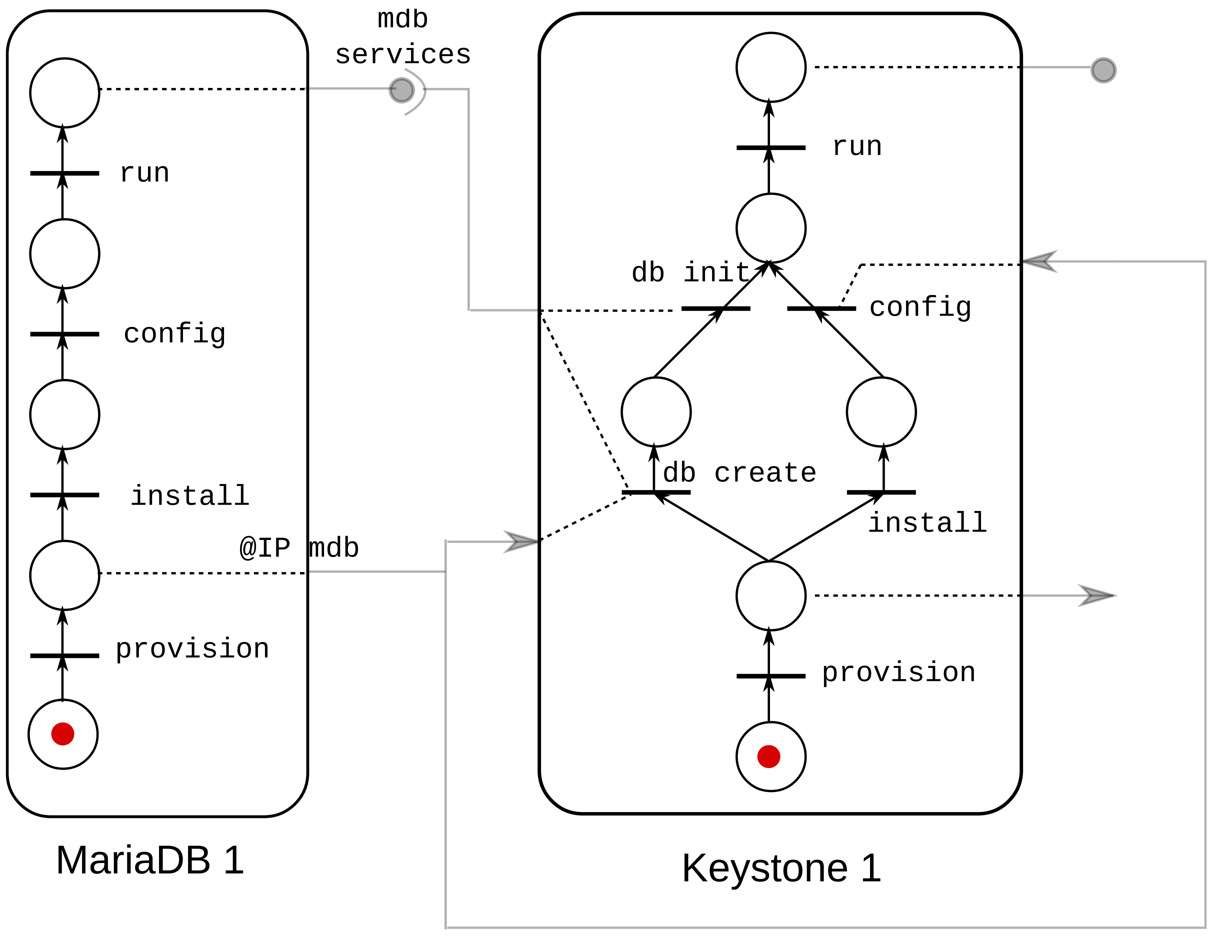


MAD COMPONENT

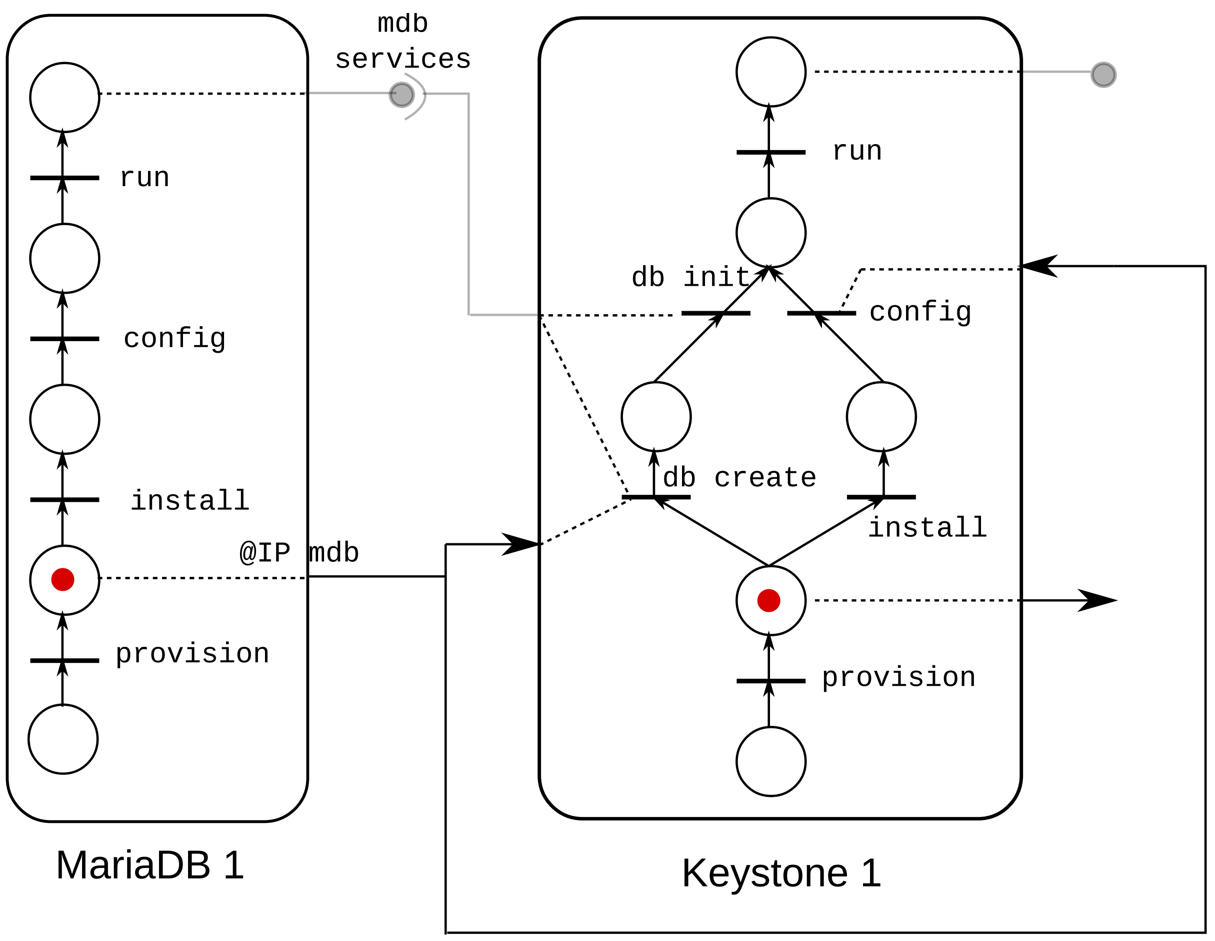


Keystone

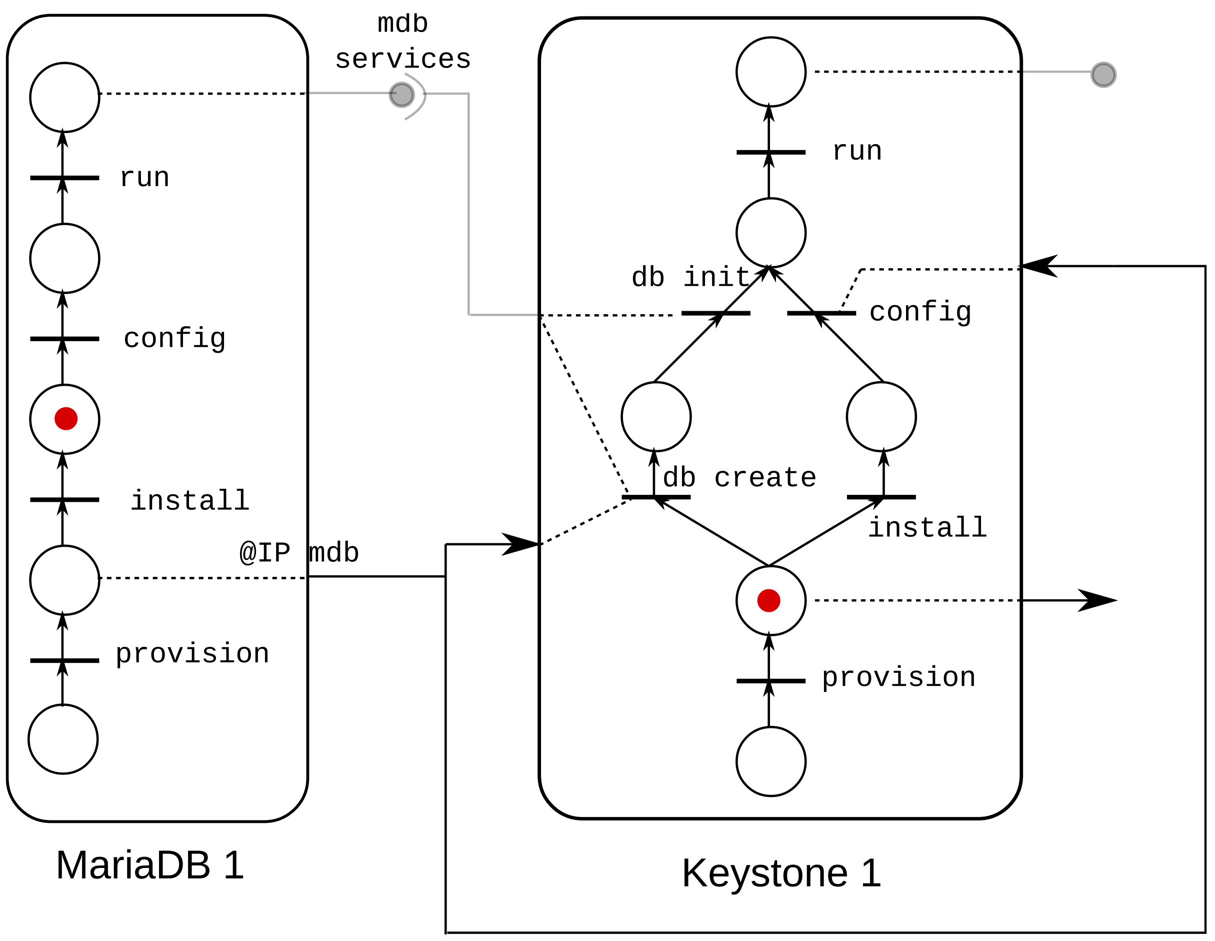
MAD ASSEMBLY



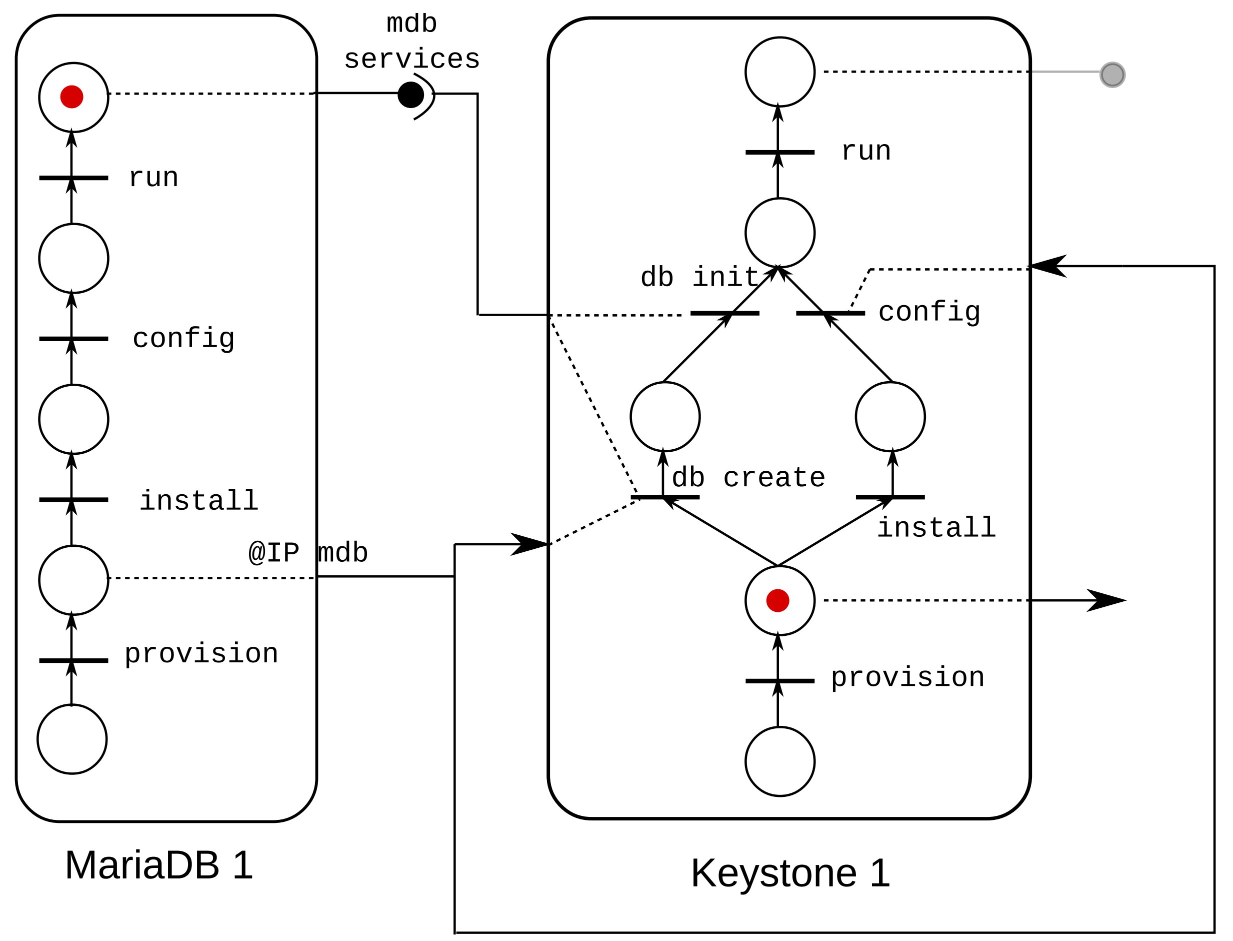
MAD ASSEMBLY



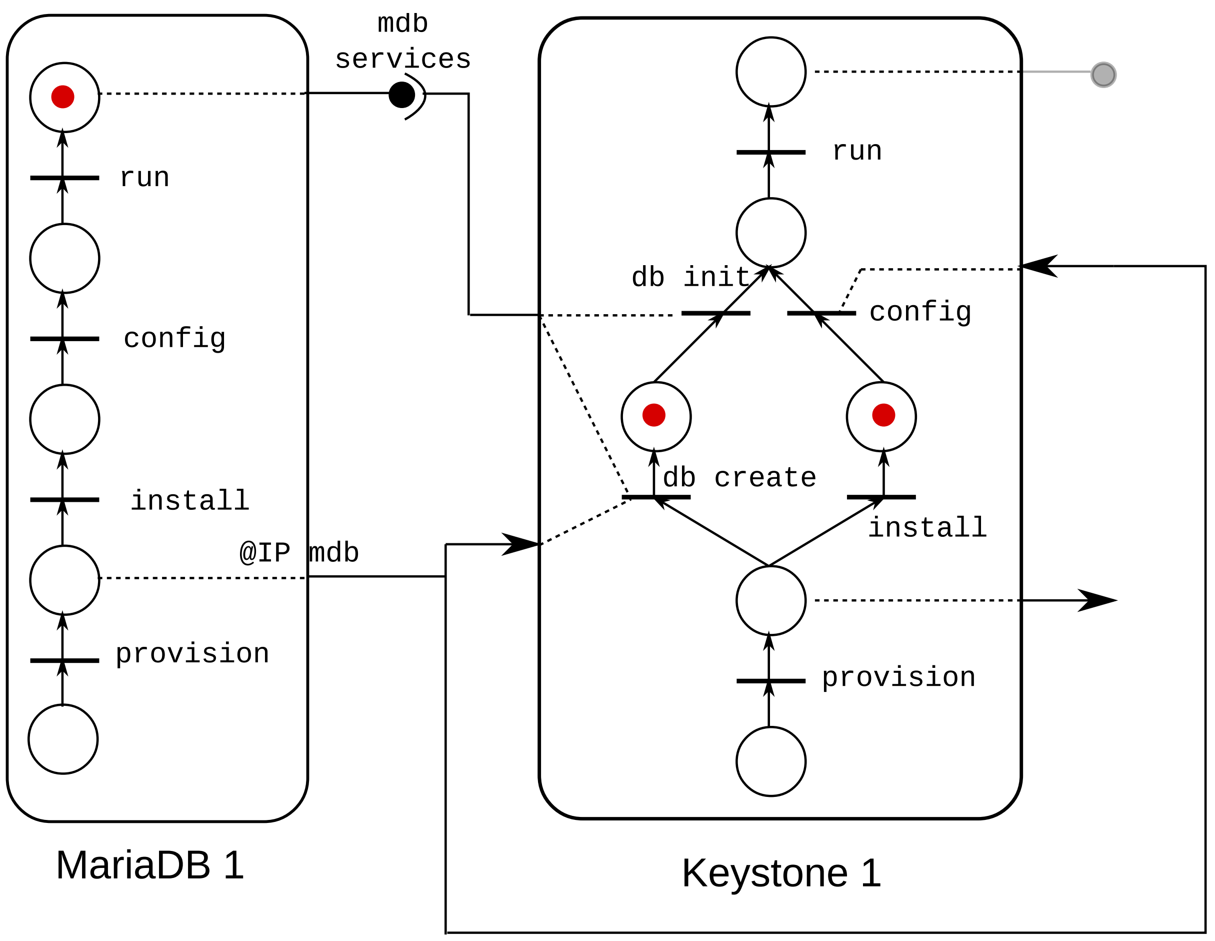
MAD ASSEMBLY



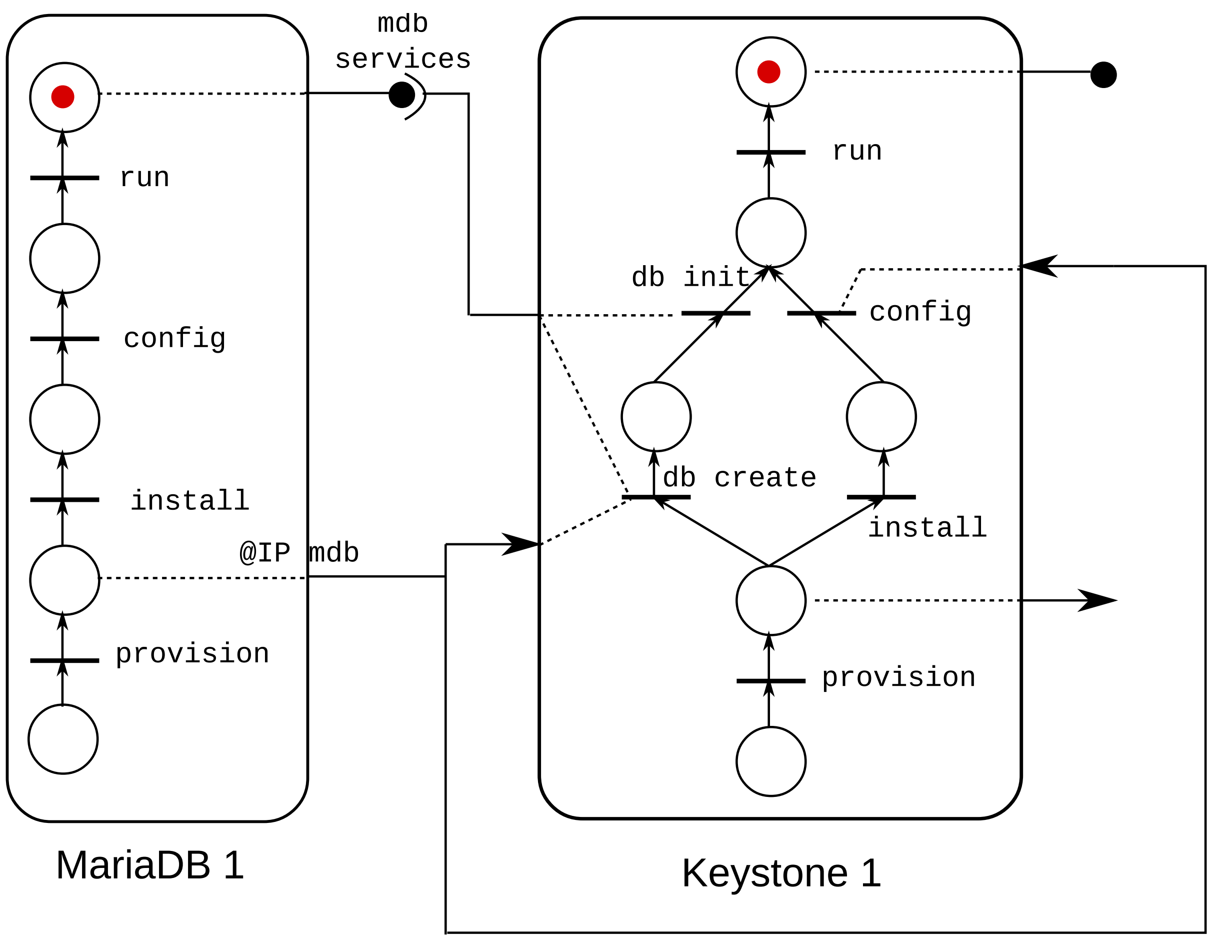
MAD ASSEMBLY



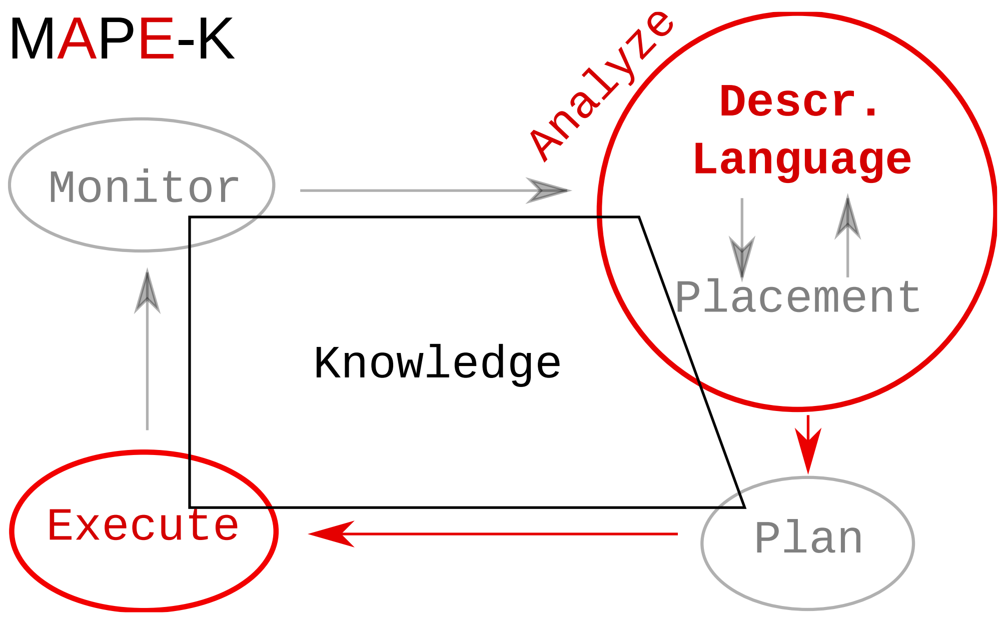
MAD ASSEMBLY



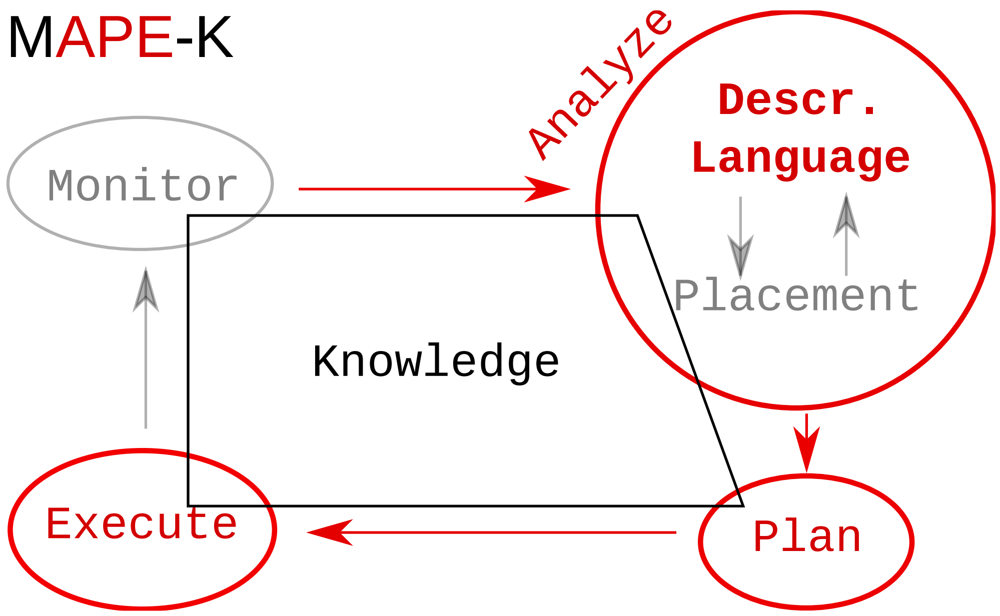
MAD ASSEMBLY



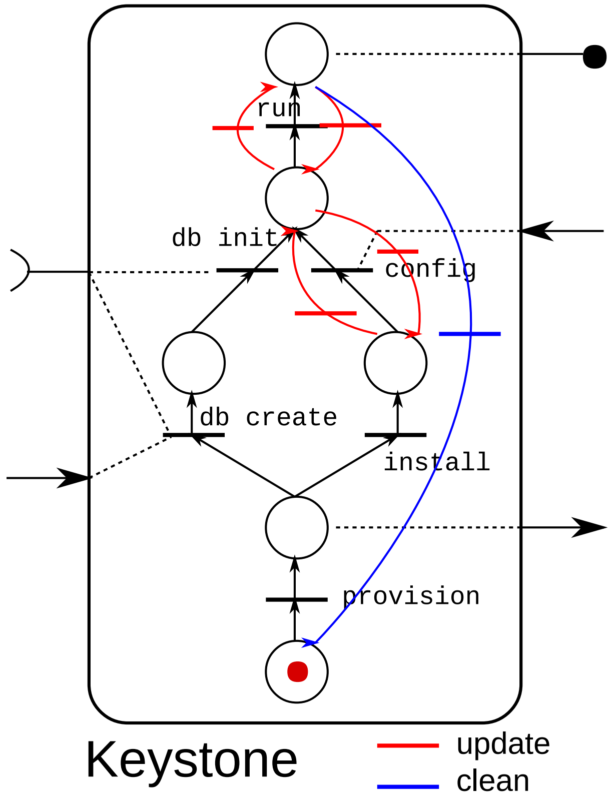
RECONFIGURATION



RECONFIGURATION



MAD COLORS


run
db init
config
db create
install
provision
Keystone
update
clean

EVALUATION

Expressivity and reliability

OpenStack use-cases

Performance

- *Deployment of OpenStack minimal (26 services)*
- *Performance comparison with Kolla and Aeolus-like*

RECONFIGURATION & CO

RECONFIGURATION

APPLICATION SIDE

- Fault Tolerance
- Automatic scaling (scale out, scale up)
- Updates (maintainability)
- Automatic software changes for various external reasons

RECONFIGURATION

INFRASTRUCTURE SIDE

- Massively distributed
- Failures
- Enter/leave
- Heterogeneity

RECONFIGURATION

CHALLENGES

- Reconfiguration expressivity
  - Go further with colors and hierarchy
- How to perform the reconfiguration?
  - relationship with the placement and monitoring
- Performances of the reconfiguration
  - decentralized reconfiguration

PERSPECTIVES

- Formalism and proofs
- Hierarchy
- Reconfiguration
- Higher abstraction level models:
  - MAD model as a back-end
  - Being more specific to applications and systems
  - Easier deployments

THANK YOU !

beyondtheclouds.github.io