

# Ariane : The digital world mapping framework - 15/06/2015 01:21 PM

---

## Abstract

Today, there is an abundance of information technologies (I.T) components that need to be mapped so that everyone can have access to full information system (I.S) mapping.

This paper presents Ariane which brings you an ubiquitous digital data pattern, allowing you to map any piece of your digital runtime in real time. Ariane is a framework that yields a highly modular system and API (Java or REST) to map any component you may have in your I.S. Once your stack is mapped, Ariane will help you to navigate the map from one point to another.

Finally Ariane can assist you to setup a new kind of I.T knowledge system with automated data miners getting the data from your living I.S and pushing them to a unified map which can be the base for a lot of new and cool analysis systems and UX features.

---

## Introduction

Information technologies (I.T) is a key enterprise factor today to acquire new clients, retain them and expand the business. Therefore the I.T engineers have developed significant amount of technologies and methodologies to acknowledge the business requests as quick as possible.

Today, business solution usually means a lot of backend applications communicating with each others to perform the necessary computing which will make the final user contented. This application stack may also depends on a lot of middleware, systems, networks and storages solutions. All these layers requires specific expertises and consequently the common pattern observed in the enterprise world is siloization (see [1] and consider I.T the same way as content).

As a result of this I.T siloization, deliveries process results in becoming more and more complex and requires a lot of coordination between the teams. This finally was bad, rather catastrophic for the time to market and so for the business (and probably business still suffers of this on many organisations). The primary actors concerned by this slow time to market was developers who were the first changing their work methodologies to Agile/XP methodologies and so speed up the releases.

But delivery was still slow as operational actors were focused on the stability of the production not speed. So a new trend emerges from this muddle to accelerate the deployment : the DevOps trend. DevOps pushed new automation systems and frameworks to the I.T (see [2],[3],[4],[5]) helping OPS improving the delivery reliability and speed. So finally OPS can be synced with the DEV as far as delivery speed is concerned, essentially helping to reduce the remoteness and the stress between these two actors.

Automation is the new big term in any organization today, a lot of money and time being spent on it. But automation doesn't act as a panacea for all the existent problems and particularly I.T expertise silos, which still prevails across the informations system. These silos still have a very bad effect on knowledge sharing and outage resolutions, which therefore can have a really bad effect on a company's credibility ([6]). And finally I.T automation introduces a new problem, that being : the

high frequency changes on information system, resulting in an impossibility to maintain I.S. Manual Human readable diagrams which are usually the first gate to help developers and operators understand the environment quickly, needs to be maintained and improved ([7]).

So now DevOps aficionados argues DevOps is not only about automation and accelerating deliveries, but is also about breaking down I.T silos and knowledge sharing ([8]) and this will be seen as the new big drift coming. But we also need to automate the knowledge base the same way we're automating the I.S delivery because as Human we need to keep the control over automated systems ([9]).

This is the primarily the mammoth target of the Ariane Project ([10]) which is a new free open source framework that helps users in writing mapping plugins to automate real time diagram generation of technologies they use every day.

---

## Ariane Architecture

In order to provide an extensible framework, Ariane is built on Eclipse Virgo ([11]) which implements OSGi standard ([12]). Such enactment helps us so that we can define a really fine grained modular system which will help us to have a maintainable system over time and clear separations of concern implementation through OSGi micro-services in a simple monolith to deploy ([13]).

Development, in a way, can be related to cooking, and as in good cooking we need good ingredients, good development needs good libraries. Therefore Ariane framework integrates additional component to Eclipse Virgo like Infinispan ([14]), Hibernate ([15]), Neo4J ([16]), Akka ([17]), Scala([18]), Groovy([19]), PrimeFaces([20]) ... These third party libraries aid in building a modern and beautiful web application.

Finally the Ariane framework provides features which are designed to be extended through Ariane plugins, primarily helping the end user to adapt the Ariane framework with its own information system environment.



Directories



---

## A. Security

The Ariane Identification Management component ([21]) is providing security service to Ariane system through Resources/Roles-Based Access Control (RRBAC) pattern ([22]). The resources, permissions and roles are stored into a basic SQL database (MariaDB ([23])) and so can be extended for any new plugins with simple SQL insert.

Currently, Users and Groups are also stored on the SQL database, thanks to Apache Shiro ([24]) we can easily provide drivers for other authentication realms through LDAP or ActiveDirectory repositories.

## B. Portal component

The Ariane Portal component ([25]) is providing extendable web services. It handles the Ariane web application context and define OSGi micro-services to expand this web application through the following registries as under :

- the main menu items registry
- the JSF bean controller registry
- the REST endpoints registry (RESTEasy ([26]))
- the JSF ([27]) views registry

Other Ariane components or plugins will just need to register their own menu entity, bean, views ... to extend the application UI. It also gives you a convenient and simple JavaServer Faces (JSF) verbs to request the Ariane IDM RBAC database in your views.

## C. Directories component

The Ariane Directories component provides ([28]) Ariane directories Java Persistence API (JPA - [29]) service to access the Ariane directories database. Ariane Directories entities are representation of your IT component definitions (like datacenters, networks, OS instances) and your organisation (applications, teams, environments).

The Ariane directories JPA service offers a convenient way to extend the entities definition in the running entity manager thanks to hibernate and OSGi (custom feature - [30]). So you can append other I.T component definitions like for example, any middleware containers you can have in your I.T to the Ariane directories.

## D. Mapping component

The Ariane Mapping component ([31]) provides Ariane mapping services relaying on Tinkerpop Blueprints ([32]), Neo4J and our ubiquitous I.T mapping model. You can inject data into the map, thanks to the mapping interface (currently memory call only) and provided REST endpoints.

It also dispenses our mapping DSL to request the map which has to be rendered in your browser throught the Ariane Mapping graph render (called Taitale - Daedalus in etruscan language) implemented with RaphaelJS ([33]).

## E. Injectors component

To help you build Ariane plugin which are sniffing your information system and inserting the topology of your I.S into Ariane Mapping database, Ariane Injectors

component ([34]) provides following tooling:

- Cache factory to create on demand Infinispan cache for you components
- Components commodities which gives you a pattern to write the I.S component data miners
- Gears you can administrate through the Ariane UI, which are handling runtime behavior of your Ariane plugin.

There are two kind of injectors :

- Internal injectors that are running into the Ariane Virgo containers and are calling the Ariane Java API (memory call yet but we will provide some distribution through Akka framework soon).
- External injectors that are running outside the Ariane Virgo containers and are calling the Ariane network API (REST and ESB through RabbitMQ([35])). They can then be written in the language of your comfort.

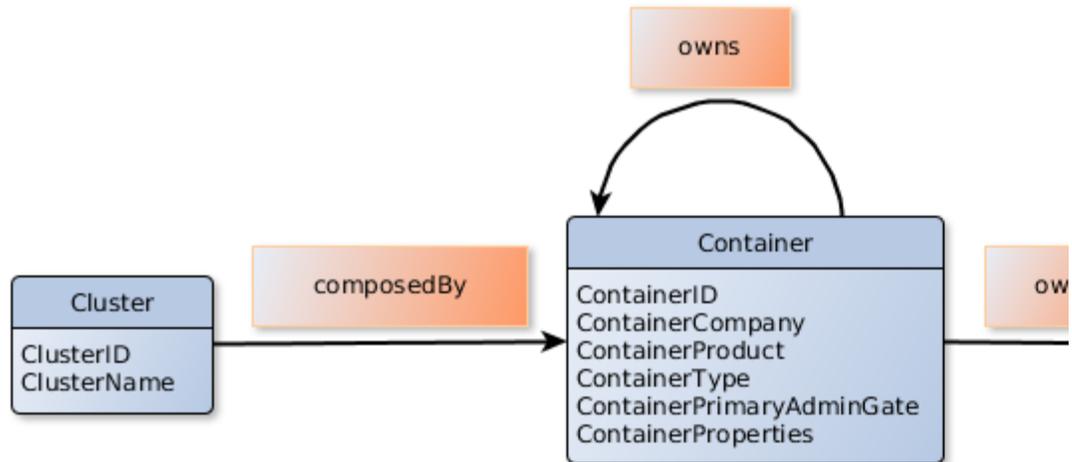
---

## MAPPING MODEL

Ariane framework's key component is its mapping component which is employed to create unified map for your information system components. It contains of micro services, that empowers us to create a unified map. Ariane mapping services rely on Neo4J which is a scalable graph database, as against the classical SQL database, thus bypassing copious amounts of table joins.

On top of classical GraphDB model (vertex and edges), Ariane provides an information system mapping model with advantages to be a simple and ubiquitous model. This model indeed has been inspired by several inter I.S communication protocols like AMQP ([36]) and it wont be an exaggeration to point that with not more than seven objects you are able to describe the most essential topology of your I.S coming from your I.S layers data (from low network technologies to highest applications technologies ([37])).

The Ariane mapping abstract domain model objects are as listed and described below :



## A. Cluster

Cluster is set of containers with unique ID which may be anything, an active/active or active/passive cluster.  
Cluster may know it's current main container in realtime.

## B. Container

Container is an administrable entity of your information system on which user can connect and authenticate. Container contains at least one administration gate which is the administration interface access. It contains one or multiple nodes which are logical entities dedicated for a specific tasks with unique ID. Container should be able to detect it's own consistent state, localization in real time (datacenter, network ....) and organisation context (developers team, support team ...)

## C. Node

Node is a logical entity devoted to specific tasks like communication, computing, storage... It consists of a set of nodes and/or endpoints which is node address URL. Node contains all necessary meta data for it's description.

## D. Gate

Gate is basically a type of node which is always a child of container but it assuredly cannot be a child of node. Gate defines services with at least one communication endpoint for remote calls. Gate must contain all necessary metadata defining the service.

## E. Endpoint

Endpoint is a node address URL. This URL is used for node to node communication (and hence, container to container). It is aware of the network stack on which it is working (for example : the network card).

## F. Link

A link is defined by two endpoints (source and destination). Link encloses its transport definition. In case of a multicast transport, link will have just the source endpoint alone.

## G. Transport

Transport is a communication protocol used by the link which can be either unidirectional or bidirectional.

---

# ARIANE MAPPING DSL

The Ariane Mapping Domain Specific Language (AMDSL - [38]) is developed with Scala to provide and ensure a fast and simple map query language, essentially based on the Ariane mapping model. User just needs to define the starting point(s), the target point(s) and the path (optional). The Mapping DSL grammar is pretty facile. It is a mixture of JSON, SQL and a pinch of Neo4J cypher (to define the path).

Basically it looks like :

```
{
  'startingPoint1': 'SQL LIKE DEFINITION'[,
  'startingPoint2': 'SQL LIKE DEFINITION'[,
  ...
  ]]
}
-[{
  'traversalPoint1': 'SQL LIKE DEFINITION'[,
  'traversalPoint2': 'SQL LIKE DEFINITION'[,
  ...
  ] | path: 'CYPHER PATH TRAVERSAL LIKE'
}]-
{
  'endPoint1': 'SQL LIKE DEFINITIONS'[,
  'endPoint2': 'SQL LIKE DEFINITIONS'[,
  ...
  ]]
}
```

Points are defined through SQL like syntax and the mapping object models. User can play with container, nodes, endpoints, repository to characterize the points they wish for their map. Path definition is not mandatory but recommended on some complex graph path traversal paths. This is because Ariane, by default, will check with all potential paths between/through the points you defined, which could consume some computing time you can reduce.

---

# ARIANE TAITALE

By using AMDSL, user can define a precise on demand map request which is then processed by the mapping back engine, which will return the list of objects found in the requested path through Ariane Mapping REST service.

The final result is then generated by the Ariane mapping graph render (known as Taitale) which compute the positioning of the returned map objects and then displays them as SVG objects. All these objects and the links between them compose the result map. Once the map is loaded, user can steer through it, observe the map objects, properties and connect to the containers they fancy for. User can edit all objects of map and export it as an image for the purpose of reports.

Maps yielded from domain specific language (DSL) are presented using two layouts :

- Network Layout : displays the network context around each containers (datacenters, routing areas, subnets) ...
- Tree Layout : a free disposition of all containers in the map with algorithm to minimize interweaving...

Finally the Ariane Taitale objects color are coming from support team color attached to container or application color which is attached to node. These colors are defined in Directories

---

## ARIANE RABBITMQ PLUGIN

RabbitMQ is an open source message-oriented middleware software which implements the Advanced Message Queuing Protocol (AMQP). RabbitMQ plugin basically extends the Ariane Core Directory and add RabbitMQ injectors intelligence into Ariane (responsible for getting the data from your RabbitMQ infrastructure, transforming them and pushing them into the Mapping DB).

Mapping RabbitMQ infrastructure will permit RabbitMQ users (IE: application developers) and operators to share running configuration in a efficient human readable manner.

### Modules

Ariane Rabbitmq Injector is divided into three submodules as listed bellow.

#### Directory

It extends the Ariane Core Directories entity by adding RabbitMQ components definitions you may deploy on your information system :

- RabbitMQ nodes where you define URL and authentication to access to the internal RabbitMQ configuration
- RabbitMQ cluster (which is just the definition of the cluster name and a RabbitMQ node set)

#### JSON Parser

This module is responsible to get the information available from the RabbitMQ node REST API. It is developed with Groovy.

#### Injector

Injector module encapsulates Gears and Components objects which are responsible to mine the data coming from the RabbitMQ components and transform them.

Each RabbitMQ injector gears are defined to interact with one data-source only to avoid multiple data-source lock into one thread. So we have the following kind of gears :

- Directory gear which is responsible to look at directory to check if there is new RabbitMQ instances (cluster or single node) to analyse. This gear is responsible to start Component gears. Period between two directories check is configurable.
- A Component gear is responsible to get configurations data of a RabbitMQ instance and push the same to *persisted* cache (Infinispan - local file) through Component object. Each component object have several fields describing the data sniffed from RabbitMQ but an important point to keep in mind here is that there are two fields for the same RabbitMQ configuration data :
  - one field for the last sniff
  - one field for the new sniff

That way we're simplifying a lot the differences algorithm between the two sniffs. Period between two sniffs is configurable. When sniff is done the injector gears are sending message to the messaging gear (through Akka actors).
- Mapping gear which is liable to get data coming from the sniffing, do the diff between last sniff and new sniff and transform the RabbitMQ configurations accordingly to fit the Messaging DB model.

## Mapping Transformations

to fit our mapping needs we transform the RabbitMQ objects into supported mapping objects.

Here is the basic mapping transformation table :

RabbitMQ Object	Mapping Object	Notes
RabbitMQ Cluster	Cluster ! 😊	
RabbitMQ Node	Container	As an administrable component RabbitMQ Node is definitively a container 😊
RabbitMQ VHost	Node (owned by a RabbitMQ Node)	<b>{container:RabbitMQ Node} -owns-&gt; {node:RabbitMQ VHost}</b>
RabbitMQ Queue	Node (owned by a RabbitMQ VHost)	<b>{node:RabbitMQ VHost} -owns-&gt; {node:RabbitMQ Queue}</b>
RabbitMQ Exchange	Node (owned by a RabbitMQ VHost)	<b>{node:RabbitMQ VHost} -owns-&gt; {node:RabbitMQ Exchange}</b>
RabbitMQ Binding	Endpoint (owned by a RabbitMQ Exchange or Queue)	<b>{node:RabbitMQ Queue} -owns-&gt; (endpoint:RabbitMQ Binding)</b>  OR  <b>{node:RabbitMQ Queue} -owns-&gt; (endpoint:RabbitMQ Binding)</b>  NOTE : queue and exchange are then linked through binding endpoints and memory link

RabbitMQ Channel + Connection	Endpoint (owned by a RabbitMQ Queue or Exchange or RabbitMQ Client Consumer or Producer)	<p>Here we can extend what we see from a RabbitMQ node to define client container / consumer and producer. We decided to merge RabbitMQ Channel + Connection as a endpoint to provide application links between these endpoints. eg :</p> <pre>[container:RabbitMQ Client process] -owns-&gt; { node:RabbitMQ Client Consumer} -owns-&gt; (end point:RabbitMQ Channel+Connection) &lt;-link-&gt; (endpoint:RabbitMQ Channel+Connection) &lt;-o wns- (node:RabbitMQ Queue)</pre> <p>OR</p> <pre>[container:RabbitMQ Client process] -owns-&gt; {node:RabbitMQ Client Publisher} -owns-&gt; (end point:RabbitMQ Channel+Connection) &lt;-link-&gt; (endpoint:RabbitMQ Channel+Connection) &lt;-o wns- {node:RabbitMQ Exchange}</pre>
-------------------------------	--	--

Finally, to link cluster node each other we are defining a specific cluster node with specific endpoints and links on each RabbitMQ node.

## FUTURE WORK

There are a lot of plugins to develop to get a complete map of the I.T world. Currently Echinopsii, the company promoting and supporting Ariane Framework can propose two plugins :

- the Ariane RabbitMQ plugin
- the Ariane TibcoRV plugin

One of our main mission is to write more plugins with the community help. Two other plugins are in the pipe and planned to be delivered this year :

- the Ariane Docker plugin
- the Ariane Cassandra plugin

Meanwhile we're working on new plugins and also improving the Ariane Core framework features. We also have many ideas to continue to improve the core framework :

- adding the temporal dimension to our map system which will then give Ariane the ability to (re) play video on what happened to the I.S.
- improve our Ariane Mapping DSL to add more intelligence behind each word pushed by the end user
- improve the Ariane Taitale graph render to give some new cool U.X features
- add some monitoring features to be merged with the Ariane On Demand map

- ....

Infrastructure System Mapping is a new exciting area of the information technology and this is really easy to find new features to add on a map like Google did with Google map... So stay tuned 😊

---

## REFERENCES

- [1] Ann Rockley and Charles Cooper. Managing enterprise content. A unified content strategy (2002). [Online] Available: <http://www.trilemon.com/wp-content/uploads/2013/01/Managing-Enterprise-Content-A-Unified-Content-Strategy.pdf>
- [2] Ben Kepes. Puppet and EMC Partner to bring DevOps to the enterprise (2014). [Online] Available: <http://www.forbes.com/sites/benkepes/2015/05/06/puppet-and-emc-partner-to-bring-devops-to-the-enterprise/>
- [3] John Minnihan. Why Docker + Ansible Make Chef and Puppet 100% Unnecessary (2014). [Online] Available : <http://devops.com/2014/06/04/using-docker-ansible/>
- [4] Corey Quinn. A taste of salt. Like Puppet, But Less Frustrating (2013) . [Online] Available : <http://blog.smartbear.com/devops/a-taste-of-salt-like-puppet-except-it-does-nt-suck/>
- [5] Ben Kepes. Chef Announces Partnerships And DevOps Workflows (2015). [Online] Available : <http://www.forbes.com/sites/benkepes/2015/04/01/chef-announces-partnerships-and-devops-workflows/>
- [6] Julie Arsenault. Reducing your incident resolution time (2014). [Online] Available : <https://www.pagerduty.com/blog/reducing-incident-resolution-time/>
- [7] Lisa Quast. Why knowledge management is important to the success of your company (2012). [Online] Available: <http://www.forbes.com/sites/lisaquast/2012/08/20/why-knowledge-management-is-important-to-the-success-of-your-company/>
- [8] John J Balena. DevOps Best Practices: Break Down Silos, Avoid Drift and Optimize for Flow (2014). [Online] Available : <http://devops.com/2014/05/16/breaking-down-silos/>
- [9] Echinopsii. Echinopsii Manifesto (2014). [Online] Available: <http://www.echinopsii.net/manifesto>
- [10] Echinopsii. Ariane project homepage (2014). [Online] Available: <http://http://ariane.echinopsii.net/>
- [11] Eclipse Foundation. Eclipse Virgo web page. [Online] Available: <https://www.eclipse.org/virgo/>
- [12] OSGi Alliance. OSGi alliance homepage. [Online] Available: <http://http://www.osgi.org/Main/HomePage>
- [13] Martin Fowler. Monolith first. (2015) [Online] Available: <http://martinfowler.com/bliki/MonolithFirst.html>
- [14] RedHat. Infinispan web site. [Online] Available: <http://infinispan.org/>
- [15] RedHat. Hibernate web site. [Online] Available: <http://hibernate.org/>
- [16] Neo Technology. Neo4J web site. [Online] Available: <http://neo4j.com/>
- [17] Typesafe. Akka web site. [Online] Available: <http://akka.io/>
- [18] EFPL. Scala web site. [Online] Available: <http://www.scala-lang.org/>
- [19] The Groovy Project. Groovy web site. [Online] Available: <http://www.groovy-lang.org/>
- [20] PrimeTek. PrimeFaces web site. [Online] Available: <http://primefaces.org/>
- [21] Echinopsii / Mathilde Ffrench. Ariane Documentation / Development Guide / Ariane Components Guide / Security (IDM). [Online] Available: <https://confluence.echinopsii.net/confluence/pages/viewpage.action?pagelD=14647453>
- [22] Les Hazlewood. The new RBAC: resource base. (2012) [Online] Available: <https://stormpath.com/blog/new-rbac-resource-based-access-control/>
- [23] MariaDB Foundation. MariaDB web site. [Online] Available: <https://mariadb.org/>
- [24] Apache Foundation. Shiro web site. [Online] Available: <http://shiro.apache.org/>
- [25] Echinopsii / Mathilde Ffrench. Ariane Documentation / Development Guide / Ariane Components Guide / Portal. [Online] Available: <https://confluence.echinopsii.net/confluence/display/AD/3+-+Portal>
- [26] RedHat. RESTEasy web page. [Online] Available: <http://restitute.jboss.org/>
- [27] Oracle. Java Server Faces technology. [Online] Available: <http://http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>
- [28] Echinopsii / Mathilde Ffrench. [Online] Available: <https://confluence.echinopsii.net/confluence/display/AD/4+-+Directory>
- [29] Oracle. Java Persistence API. [Online] Available: <http://www.oracle.com/technetw>

ork/java/javaaa/tech/persistence-jsp-140049.html  
[30] Echinopsii / GitHub branch of Hibernate ORM allowing extension of EntityManager entities set at runtime with OSGi. [Online] Available: <https://github.com/echinopsii/net.echinopsii.3rdparty.hibernate-orm>  
[31] Echinopsii / Mathilde Ffrench. Ariane Documentation / Development Guide / Ariane Components Guide / Mapping. [Online] Available: <https://confluence.echinopsii.net/confluence/display/AD/5+-+Mapping>  
[32] Tinkerpop. Tinkerpop Blueprints home page. [Online] Available: <https://github.com/tinkerpop/blueprints/wiki>  
[33] Dimitri Baranovskiy. Raphael javascript library homepage. [Online] Available: <http://http://raphaeljs.com/>  
[34] Echinopsii / Mathilde Ffrench. Ariane Documentation / Development Guide / Ariane Components Guide / Injector. [Online] Available: <https://confluence.echinopsii.net/confluence/display/AD/6+-+Injector>  
[35] Pivotal. RabbitMQ homepage. [Online] Available: <https://www.rabbitmq.com/>  
[36] OASIS. AMQP homepage. [Online] Available: <https://www.amqp.org/>  
[37] Echinopsii / Mathilde Ffrench. Ariane Documentation / Development Guide / Ariane Components Guide / Mapping / Abstract domain model. [Online] Available: <https://confluence.echinopsii.net/confluence/display/AD/52+-+Abstract+domain+model>  
[38] Echinopsii / Mathilde Ffrench. Ariane Documentation / Development Guide / Ariane Components Guide / Mapping / Mapping DSL (mdsl). [Online] Available: <https://confluence.echinopsii.net/confluence/pages/viewpage.action?pageId=10158150>

---

This work is licensed under a [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/).

actors apm ariane burst complexity devops fractal graph graphdb itoa itsm  
karaf mapping monitoring neo4j osgi primefaces virgo **visualizati**  
**on**