

OCCI Monitoring

Extending the OCCI API with monitoring capabilities

Augusto Ciuffoletti

Dept. of Computer Science - Univ. of Pisa

April 10, 2013

- ▶ A first step towards SLA

- ▶ A first step towards SLA
- ▶ Give the user an interface to arrange a monitoring infrastructure

- ▶ A first step towards SLA
- ▶ Give the user an interface to arrange a monitoring infrastructure
- ▶ Useful especially in the case the user is in its turn a service provider...

- ▶ A first step towards SLA
- ▶ Give the user an interface to arrange a monitoring infrastructure
- ▶ Useful especially in the case the user is in its turn a service provider...
- ▶ ... or in the case the user wants to "double check" the provided service level

- ▶ A first step towards SLA
- ▶ Give the user an interface to arrange a monitoring infrastructure
- ▶ Useful especially in the case the user is in its turn a service provider...
- ▶ ... or in the case the user wants to "double check" the provided service level
- ▶ In the former case the user uses the monitoring infrastructure to ensure quality of service...

- ▶ A first step towards SLA
- ▶ Give the user an interface to arrange a monitoring infrastructure
- ▶ Useful especially in the case the user is in its turn a service provider...
- ▶ ... or in the case the user wants to "double check" the provided service level
- ▶ In the former case the user uses the monitoring infrastructure to ensure quality of service...
- ▶ ...and for billing purposes

- ▶ A first step towards SLA
- ▶ Give the user an interface to arrange a monitoring infrastructure
- ▶ Useful especially in the case the user is in its turn a service provider...
- ▶ ... or in the case the user wants to "double check" the provided service level
- ▶ In the former case the user uses the monitoring infrastructure to ensure quality of service...
- ▶ ...and for billing purposes
- ▶ Simple and optional, aligned with OCCI

- ▶ A first step towards SLA
- ▶ Give the user an interface to arrange a monitoring infrastructure
- ▶ Useful especially in the case the user is in its turn a service provider...
- ▶ ... or in the case the user wants to "double check" the provided service level
- ▶ In the former case the user uses the monitoring infrastructure to ensure quality of service...
- ▶ ...and for billing purposes
- ▶ Simple and optional, aligned with OCCI
- ▶ Two types: the Collector and the Sensor

- ▶ The Collector is a Link, in the OCCI terminology;

- ▶ The Collector is a Link, in the OCCI terminology;
- ▶ It has two distinct roles:

- ▶ The Collector is a Link, in the OCCI terminology;
- ▶ It has two distinct roles:
 - ▶ extract operational parameters from the Source resource

- ▶ The Collector is a Link, in the OCCI terminology;
- ▶ It has two distinct roles:
 - ▶ extract operational parameters from the Source resource
 - ▶ deliver such parameters to Target resource

- ▶ The Collector is a Link, in the OCCI terminology;
- ▶ It has two distinct roles:
 - ▶ extract operational parameters from the Source resource
 - ▶ deliver such parameters to Target resource
- ▶ There are innumerable options for both roles:

- ▶ The Collector is a Link, in the OCCI terminology;
- ▶ It has two distinct roles:
 - ▶ extract operational parameters from the Source resource
 - ▶ deliver such parameters to Target resource
- ▶ There are innumerable options for both roles:
 - ▶ the operational parameters are as many as the types of resources (and more)

- ▶ The Collector is a Link, in the OCCI terminology;
- ▶ It has two distinct roles:
 - ▶ extract operational parameters from the Source resource
 - ▶ deliver such parameters to Target resource
- ▶ There are innumerable options for both roles:
 - ▶ the operational parameters are as many as the types of resources (and more)
 - ▶ the transport media are also extremely variable: tcp connection, push/pop, database, sms etc.

- ▶ The Collector is a Link, in the OC CI terminology;
- ▶ It has two distinct roles:
 - ▶ extract operational parameters from the Source resource
 - ▶ deliver such parameters to Target resource
- ▶ There are innumerable options for both roles:
 - ▶ the operational parameters are as many as the types of resources (and more)
 - ▶ the transport media are also extremely variable: tcp connection, push/pop, database, sms etc.
- ▶ The OC CI way: provide plugin extensions

- ▶ The Collector is a Link, in the OCCI terminology;
- ▶ It has two distinct roles:
 - ▶ extract operational parameters from the Source resource
 - ▶ deliver such parameters to Target resource
- ▶ There are innumerable options for both roles:
 - ▶ the operational parameters are as many as the types of resources (and more)
 - ▶ the transport media are also extremely variable: tcp connection, push/pop, database, sms etc.
- ▶ The OCCI way: provide plugin extensions
- ▶ Plugins are organized into two collections: ToolSet and CollectorSet

- ▶ The Sensor is a Resource in the OCCI terminology

- ▶ The Sensor is a Resource in the OCCI terminology
- ▶ It is specific for monitoring: its role is to process or aggregate the output of one or more Collectors

- ▶ The Sensor is a Resource in the OCCI terminology
- ▶ It is specific for monitoring: its role is to process or aggregate the output of one or more Collectors
- ▶ Also in this case, the ways to aggregate and process monitoring data are too many to envision an index

- ▶ The Sensor is a Resource in the OCCI terminology
- ▶ It is specific for monitoring: its role is to process or aggregate the output of one or more Collectors
- ▶ Also in this case, the ways to aggregate and process monitoring data are too many to envision an index
- ▶ For instance filtering, interpolation, combination

- ▶ The Sensor is a Resource in the OCCI terminology
- ▶ It is specific for monitoring: its role is to process or aggregate the output of one or more Collectors
- ▶ Also in this case, the ways to aggregate and process monitoring data are too many to envision an index
- ▶ For instance filtering, interpolation, combination
- ▶ For this reason a Sensor can be further specified using mixins in the AggregatorSet collection

- ▶ The Sensor is a Resource in the OCCI terminology
- ▶ It is specific for monitoring: its role is to process or aggregate the output of one or more Collectors
- ▶ Also in this case, the ways to aggregate and process monitoring data are too many to envision an index
- ▶ For instance filtering, interpolation, combination
- ▶ For this reason a Sensor can be further specified using mixins in the AggregatorSet collection
- ▶ The Sensor delivers its results to another Resource through a Collector

- ▶ The Sensor is a Resource in the OCCI terminology
- ▶ It is specific for monitoring: its role is to process or aggregate the output of one or more Collectors
- ▶ Also in this case, the ways to aggregate and process monitoring data are too many to envision an index
- ▶ For instance filtering, interpolation, combination
- ▶ For this reason a Sensor can be further specified using mixins in the AggregatorSet collection
- ▶ The Sensor delivers its results to another Resource through a Collector
- ▶ The target resource can be, for instance, a Compute Resource that implements a resource management strategy

- ▶ The mixins collections are characterized by constrained attributes

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done
- ▶ CollectorSet (Collector only):

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done
- ▶ CollectorSet (Collector only):
 - ▶ Input: reference to source Sensor attributes

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done
- ▶ CollectorSet (Collector only):
 - ▶ Input: reference to source Sensor attributes
 - ▶ Control: how measurements are published

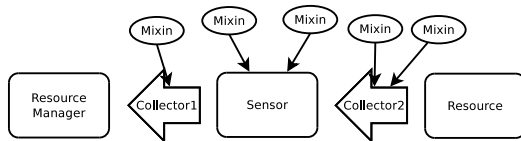
- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done
- ▶ CollectorSet (Collector only):
 - ▶ Input: reference to source Sensor attributes
 - ▶ Control: how measurements are published
- ▶ AggregatorSet (Sensor only):

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done
- ▶ CollectorSet (Collector only):
 - ▶ Input: reference to source Sensor attributes
 - ▶ Control: how measurements are published
- ▶ AggregatorSet (Sensor only):
 - ▶ Input: reference to output Collector attributes

- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done
- ▶ CollectorSet (Collector only):
 - ▶ Input: reference to source Sensor attributes
 - ▶ Control: how measurements are published
- ▶ AggregatorSet (Sensor only):
 - ▶ Input: reference to output Collector attributes
 - ▶ Control: how

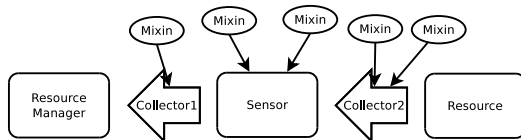
- ▶ The mixins collections are characterized by constrained attributes
- ▶ ToolSet (Collector only):
 - ▶ Metric Attributes: measured metrics
 - ▶ Control attributes: how measurements are done
- ▶ CollectorSet (Collector only):
 - ▶ Input: reference to source Sensor attributes
 - ▶ Control: how measurements are published
- ▶ AggregatorSet (Sensor only):
 - ▶ Input: reference to output Collector attributes
 - ▶ Control: how
 - ▶ Metric: what

A single stage scenario



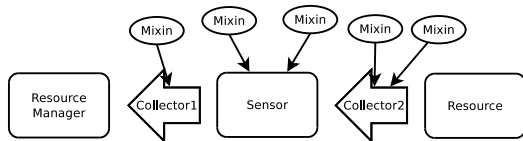
- This corresponds to the basic case

A single stage scenario



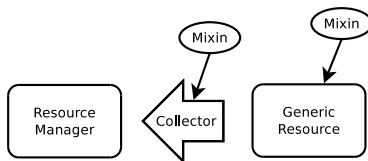
- ▶ This corresponds to the basic case
- ▶ The collectors are characterized with two ToolSet mixins (Collector2),

A single stage scenario



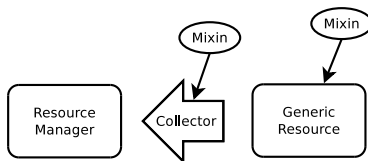
- ▶ This corresponds to the basic case
- ▶ The collectors are characterized with two ToolSet mixins (Collector2),
- ▶ and one CollectorSet mixin (Collector1)

A self-monitoring resource



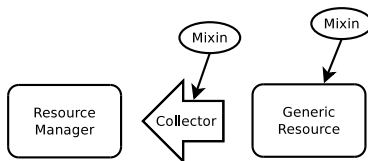
- This is the simplest case

A self-monitoring resource



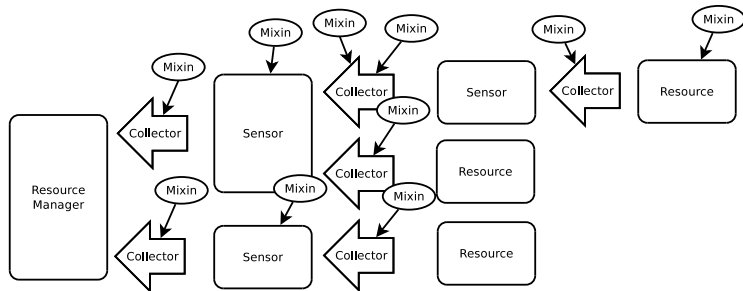
- ▶ This is the simplest case
- ▶ The monitored resource has `AggregatorSet` mixins that expose metrics and controls

A self-monitoring resource



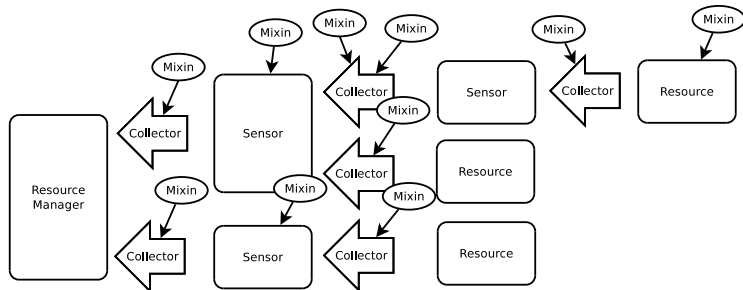
- ▶ This is the simplest case
- ▶ The monitored resource has `AggregatorSet` mixins that expose metrics and controls
- ▶ e.g.: A Compute Resource with a *syslog* processor as a mixin

A multi-stage monitoring infrastructure



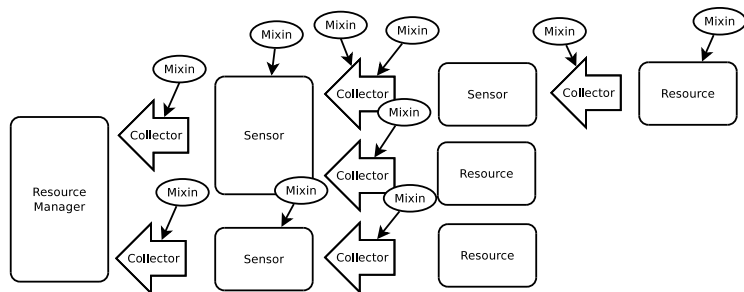
- An example to show how generic is the model

A multi-stage monitoring infrastructure



- ▶ An example to show how generic is the model
- ▶ Multistage sensors: useful to cross provider boundaries

A multi-stage monitoring infrastructure



- ▶ An example to show how generic is the model
- ▶ Multistage sensors: useful to cross provider boundaries
- ▶ Combining measurements: a metric can result from combination

- ▶ Conformance profiles: to accomodate the presence of providers that do not implement a monitoring interface

Also in the document

- ▶ Conformance profiles: to accomodate the presence of providers that do not implement a monitoring interface
- ▶ Security issues

Also in the document

- ▶ Conformance profiles: to accomodate the presence of providers that do not implement a monitoring interface
- ▶ Security issues
- ▶ A detailed example using the http rendering

Also in the document

- ▶ Conformance profiles: to accomodate the presence of providers that do not implement a monitoring interface
- ▶ Security issues
- ▶ A detailed example using the http rendering
- ▶ ...and several bugs.

Also in the document

- ▶ Conformance profiles: to accomodate the presence of providers that do not implement a monitoring interface
- ▶ Security issues
- ▶ A detailed example using the http rendering
- ▶ ...and several bugs.

That's all...