OCCI Monitoring

Extending the OCCI API with monitoring capabilities

Augusto Ciuffoletti

Dept. of Computer Science - Univ. of Pisa

April 11, 2013

Motivation

- A first step towards SLA

Motivation

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

Motivation

- 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...

Motivation

- 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

Motivation

- 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...

Motivation

- 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

Motivation

- 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

Motivation

- 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

Basics: the Collector

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

Basics: the Collector

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

Basics: the Collector

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

Basics: the Collector

- 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

Basics: the Collector

- 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:

Basics: the Collector

- 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)

Basics: the Collector

- 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 trasport media are also extremely variable: tcp connection, push/pop, database, sms etc.

Basics: the Collector

- 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 trasport media are also extremely variable: tcp connection, push/pop, database, sms etc.
- The OCCI way: provide plugin extensions

Basics: the Collector

- 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 trasport 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

Basics: the Sensor

- The Sensor is a Resource in the OCCI terminology

Basics: the Sensor

- 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

Basics: the Sensor

- 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

Basics: the Sensor

- 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

Basics: the Sensor

- 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

Basics: the Sensor

- 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

Basics: the Sensor

- 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 mix-ins: constraints

- The mixins collections are characterized by constrained attributes

The mix-ins: constraints

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

The mix-ins: constraints

- The mixins collections are characterized by constrained attributes
- ToolSet (Collector only):
  - Metric Attributes: the name corresponds to a measured metrics

The mix-ins: constraints

- The mixins collections are characterized by constrained attributes
- ToolSet (Collector only):
  - Metric Attributes: the name corresponds to a measured metrics
  - Control attributes: how measurements are done

The mix-ins: constraints

- The mixins collections are characterized by constrained attributes
- ToolSet (Collector only):
  - Metric Attributes: the name corresponds to a measured metrics
  - Control attributes: how measurements are done
- PublisherSet (Sensor only):

The mix-ins: constraints

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

The mix-ins: constraints

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

The mix-ins: constraints

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

The mix-ins: constraints

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

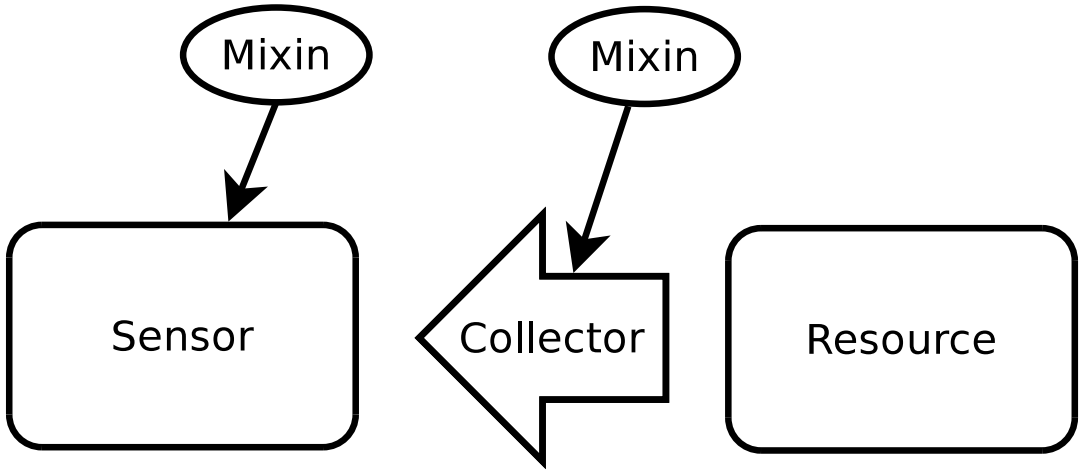
The mix-ins: constraints

- The mixins collections are characterized by constrained attributes
- ToolSet (Collector only):
  - Metric Attributes: the name corresponds to a measured metrics
  - Control attributes: how measurements are done
- PublisherSet (Sensor only):
  - Input: reference to source Sensor attributes
  - Control: how measurements are published
- AggregatorSet (Sensor only):
  - Input: reference to output Collector attributes
  - Control: how the specific aggregation is done

The mix-ins: constraints

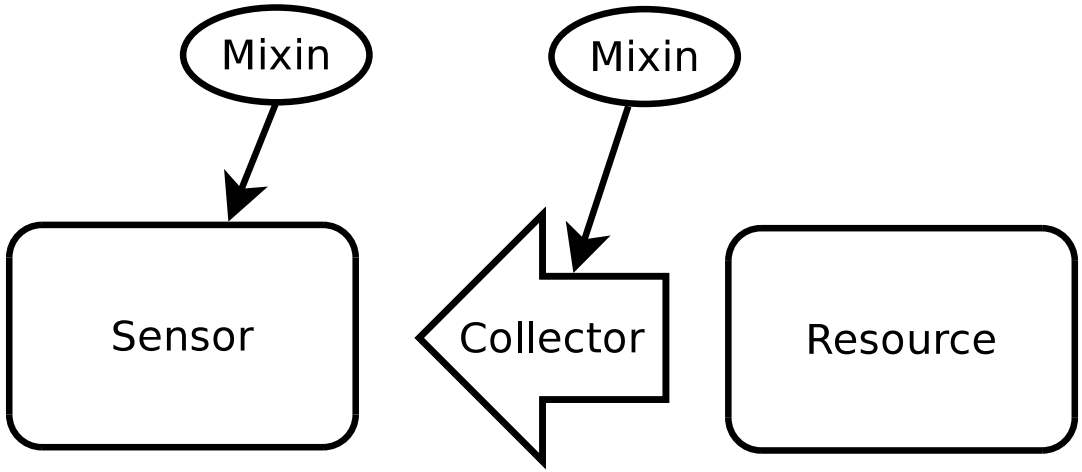
- The mixins collections are characterized by constrained attributes
- ToolSet (Collector only):
  - Metric Attributes: the name corresponds to a measured metrics
  - Control attributes: how measurements are done
- PublisherSet (Sensor only):
  - Input: reference to source Sensor attributes
  - Control: how measurements are published
- AggregatorSet (Sensor only):
  - Input: reference to output Collector attributes
  - Control: how the specific aggregation is done
  - Metric: the name corresponds to an aggregated metric

A single stage scenario



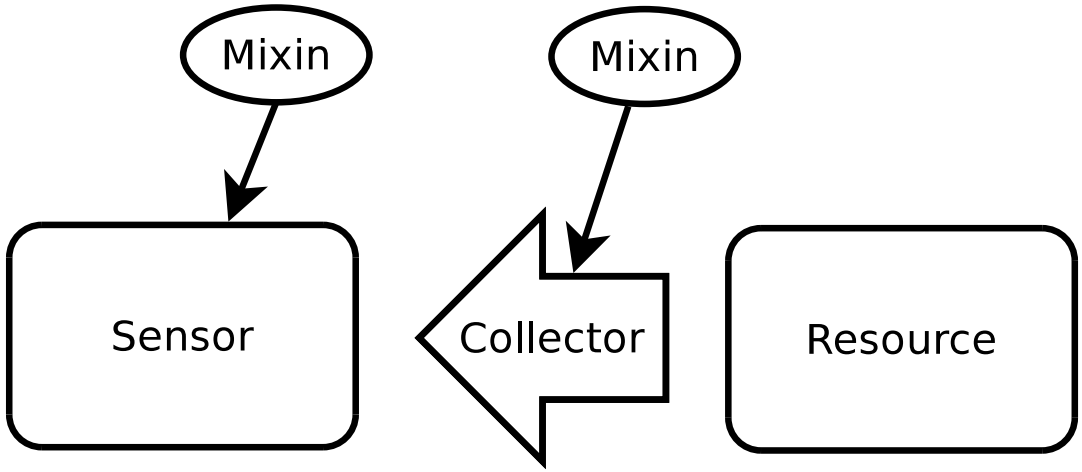
- This corresponds to the basic case:

A single stage scenario



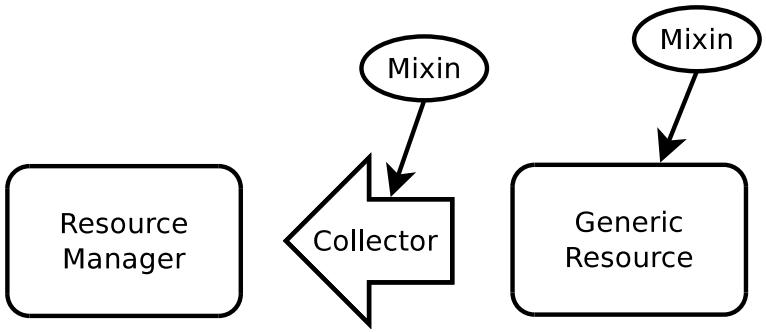
- This corresponds to the basic case:
- The collector is characterized with a ToolSet mixin,

A single stage scenario



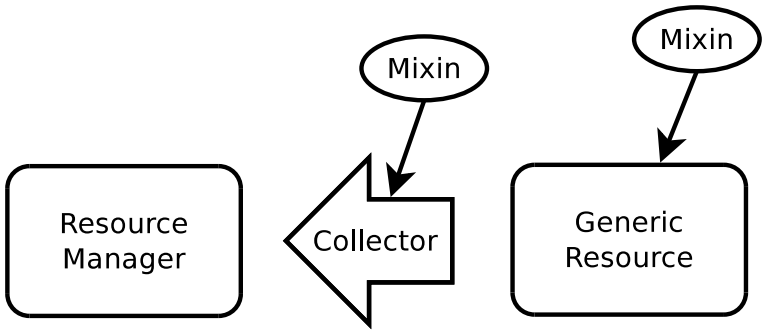
- This corresponds to the basic case:
- The collector is characterized with a ToolSet mixin,
- while the sensor has a PublisherSet attribute.

A self-monitoring resource



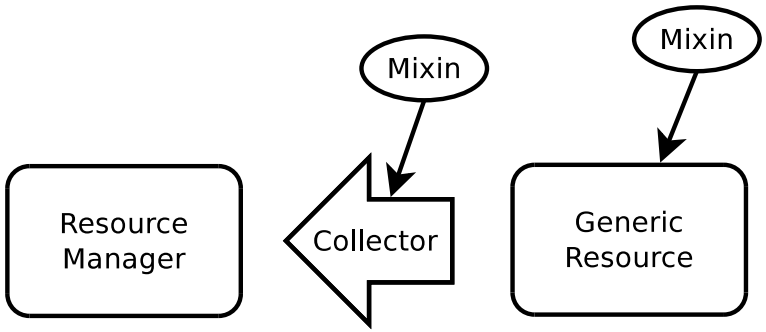
- This is the simplest case

A self-monitoring resource



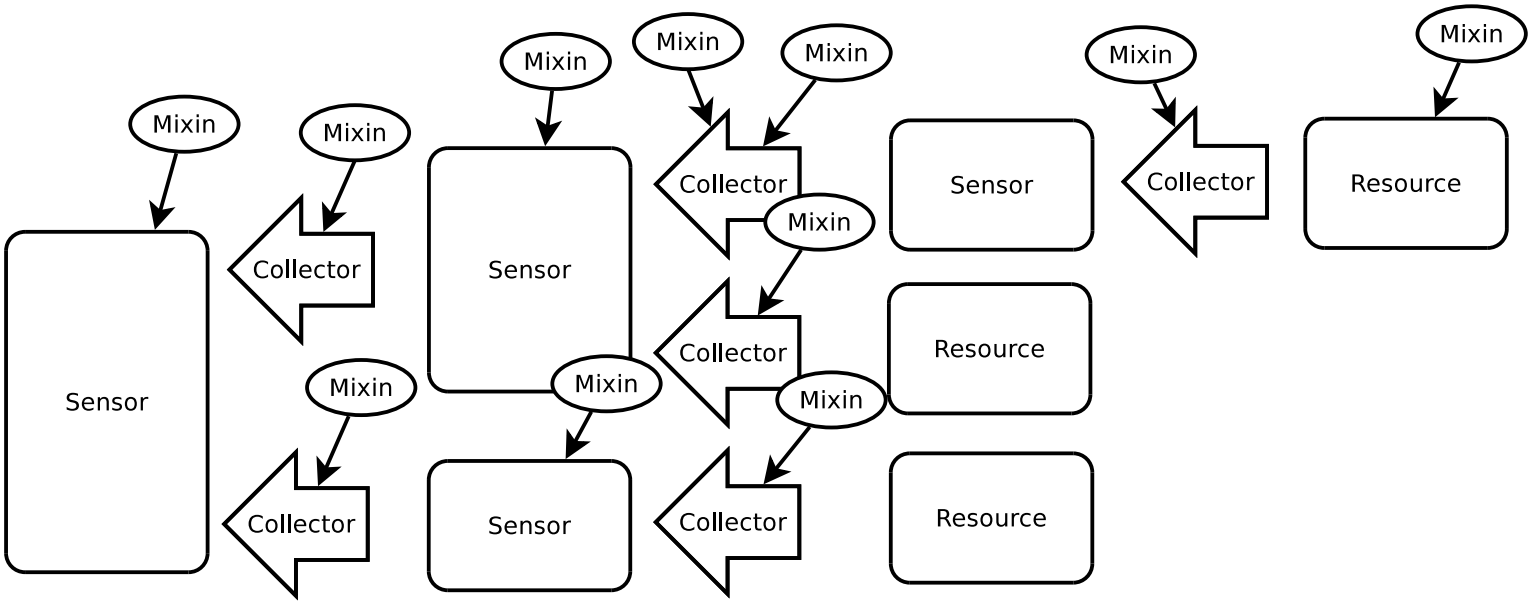
- This is the simplest case
- The monitored resource has AggregatorSet mixins that expose metrics and controls

A self-monitoring resource



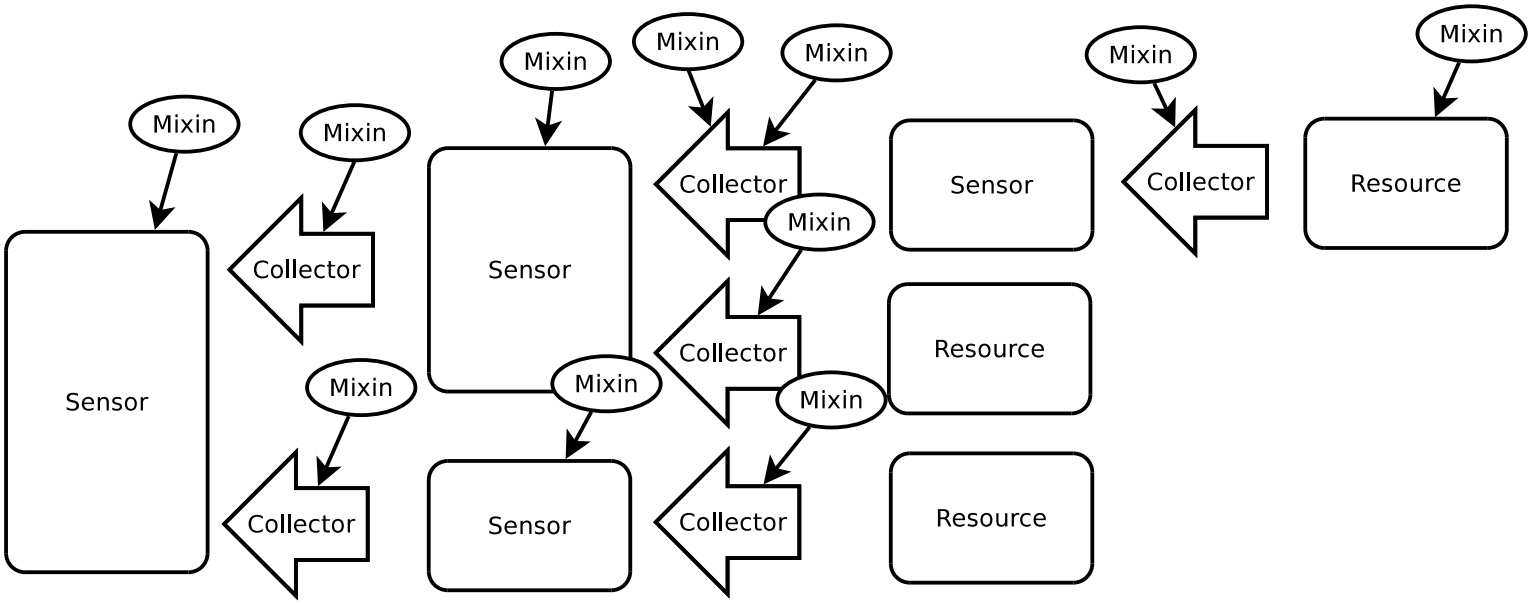
- This is the simplest case
- The monitored resource has AggregatorSet mixins that expose metrics and controls
- e.g.: A Compute Resource with a *syslog* processor as a mixin

A multi-stage monitoring infrastructure



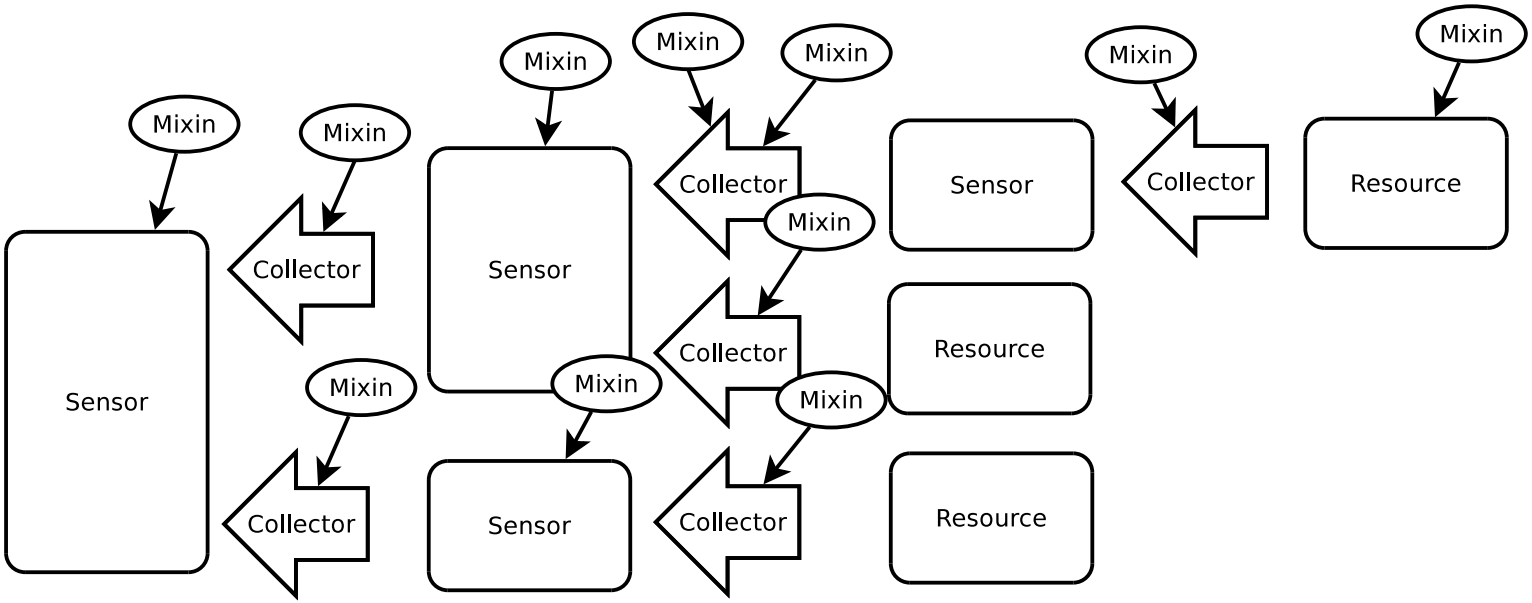
- An example to show how generic is the model

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

Also in the document

- 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...