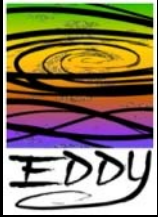


CER Factory

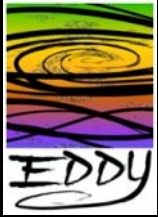
A simple service for the
manufacturing EDDY CERs

Chas DiFatta (chas@cmu.edu)
Jim Gargani (jgargani@cmu.edu)
Kevin Miller (kcmiller@duke.edu)
Mark Poepping (poepping@cmu.edu)



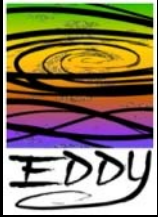
Concept

- Facility to easily create CERs from a limited set of event attributes
- Simplify using the backplane for non-Java developers
- Reduce the barrier to adoption for embedded system developers



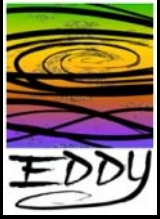
Use Case Profiles

- **Experimenter:**
 - Why: Initial testing the backplane
 - Interface: from Perl, C, etc. to CER factory service
- **Developer:**
 - Why: rapid development
 - Interface: from Perl, C, etc. to CER factory service
- **Embedded systems:**
 - When normalizer cannot fit on system
 - Interface: send factory template directly to CER factory service



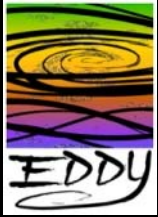
Goals

- Minimal variables required by factory to create an official CER
 - Name of event
 - Data
- Factory will make best attempt to populate the rest of the CER

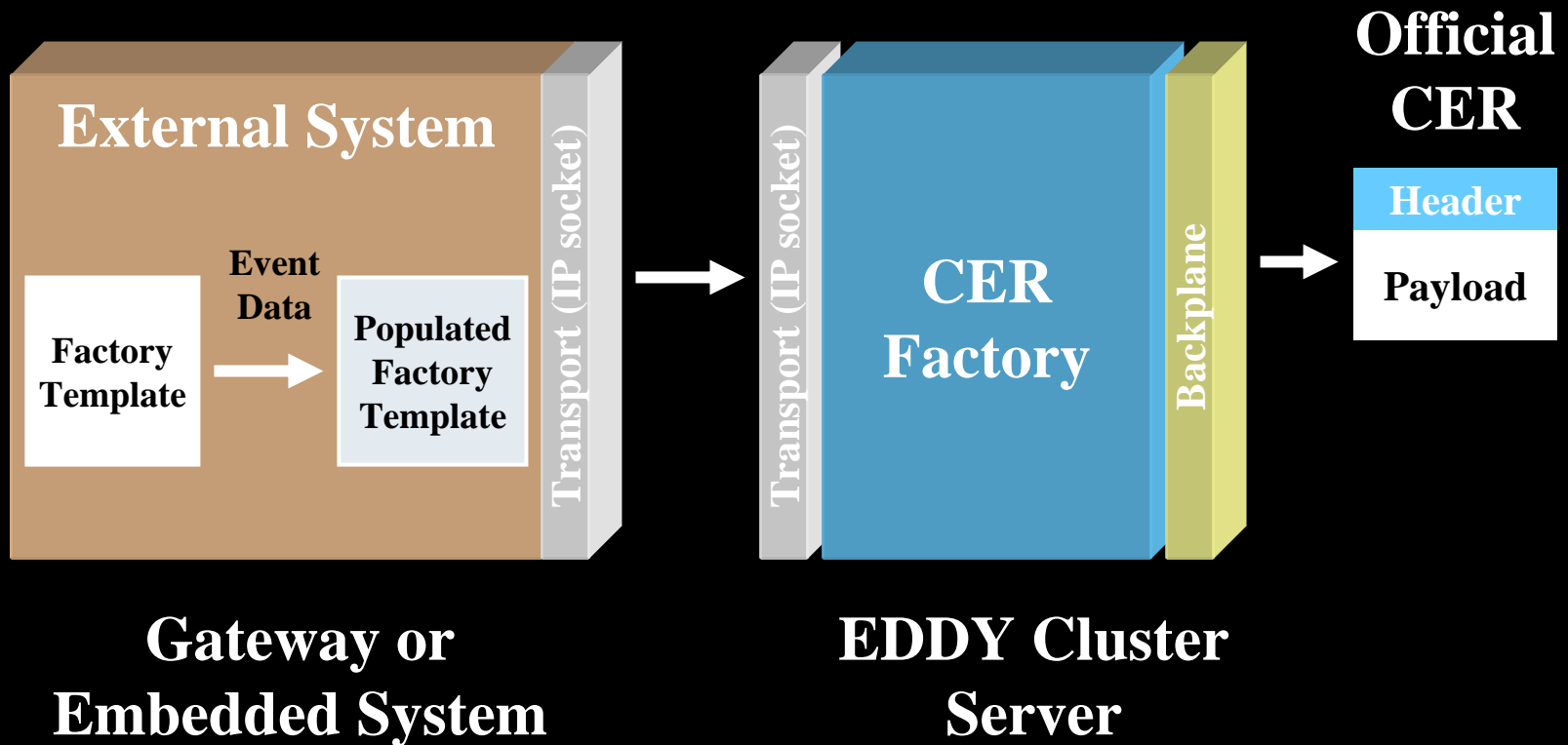


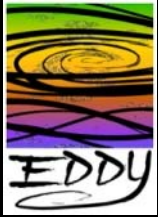
Factory Objectives

- Attach directly to EDDY backplane
- Simple push of a template
- Flexible transport and authentication:
 - http(s), SSL, SOAP, Native (socket)
- Configurable authentication for simple devices
 - Using shared keys can be an option



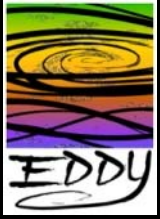
Basic Architecture





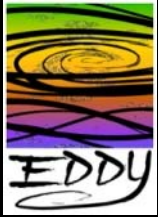
Possible Factory Transports and Authentication

- SSL (initial candidate)
 - certificate
- Unix Domain Socket (initial candidate)
 - Unix primitives
- Service (well known TCP port)
 - TCP wrappers and/or shared keys and encrypt payload
- SOAP
 - certificate
- ActiveMQ
 - certificate
- Native EDDY
 - certificate



CER Factory Requirements

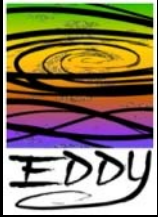
- Transport will accommodate SSL and Unix domain sockets initially but will support plug-in transport architecture.
 - Initially, transport architecture will only accommodate one transport plug-in at a time
- Factory will not report back to client if template contains an error, but will log as an internal error



CER Factory

Server Responsibilities

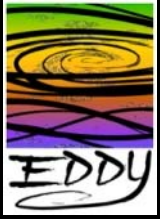
- Validate essential CER elements as a proper CER Factory document
- Create an official CER from the CER Factory Template
 - Auto-populate default fields
 - Route on CER Backplane



CER Factory

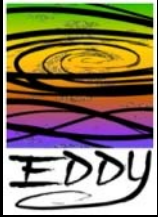
Client Responsibilities

- Define essential header elements required template fields
- Define additional optional CER elements as needed
 - Additional CER elements must allow generation of valid official CER
- Transport (SSL and Unix domain sockets initially)
- No libraries required on client



CER Factory Template

- Flattened version of official CER
- Structure of official CER is removed
- A minimal set of CER elements are required relying on Factory to supply defaults
- Factory Template supports the inclusion of any elements that comprise an official CER
- Factory Template itself is an XML schema

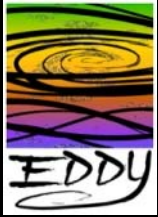


CER Factory

Template Example

- The following shows a Raw Argus network event as a Factory Template:

```
<?xml version="1.0" encoding="UTF-8"?>
<cerFactory>
  <eventInfo.oid>1001</eventInfo.oid> <!-- Argus -->
  <eventInfo.eventHostname>argus1.net.cmu.edu</eventInfo.eventHostname>
  <eventInfo.eventClass>1</eventInfo.eventClass> <!-- Network -->
  <eventInfo.warningLevel>7</eventInfo.warningLevel> <!-- Informational -->
  <dataPayload.payloadType>1</dataPayload.payloadType> <!-- Raw -->
  <dataPayload.payload>
    ASAA7ACACAB/AAABAAABcAFIAAAAAAF5RDRO6wAGTUVENE7rAAZNRcCoAQHv
    //6EQARNQdsitYAAAAABAAAAAAAAAAEAAAFqAAABQAAAAAAAAAAAAAAAAAAAgQ
    CAAADYgr27kBAF5//pCIQAATk9USUZZICogSFRUUC8xLjENCkhPU1Q6IDIz
    OS4yNTUuMjU1LjI1MDoxOTAwDQpDQUNIRS1DT05UUk9MOiBtYXgtYWdlPTEy
    MA0KTE9DQVRJT046IGh0dHA6Ly8xOTIuMTY4LjEuMTU1LjU1LjU1LjU1LjU1LjU1
    Ck5UOiB1cm46c2M=
  </dataPayload.payload>
</cerFactory>
```

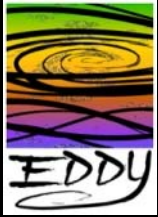


CER Factory

Template Example

- The following shows a network SNMP event (using the userTag) as a Factory Template:

```
<?xml version="1.0" encoding="UTF-8"?>
<cerFactory>
  <eventInfo.oid>1010</eventInfo.oid> <!-- Network Key/Value -->
  <eventInfo.eventHostname>switch1.net.cmu.edu</eventInfo.eventHostname>
  <eventInfo.eventClass>1</eventInfo.eventClass> <!-- Network -->
  <eventInfo.warningLevel>1</eventInfo.warningLevel> <!-- Informational-->
  <eventInfo.userTag> <!-- SNMP MIBS -->
    <key>interfaces.ifTable.ifEntry.ifInOctets.20</key>
    <value>1520103453</value>
    <key>interfaces.ifTable.ifEntry.ifOperStatus.25</key>
    <value>up(1)</value>
  </eventInfo.userTag>
  <dataPayload.payloadType>1</dataPayload.payloadType> <!-- Raw -->
  <dataPayload.payload/> <!-- No Payload -->
</cerFactory>
```



CER Factory Template Example

The following shows an environmental heat sensor event as a Factory Template:

```
<?xml version="1.0" encoding="UTF-8"?>
<cerFactory>
  <eventInfo.oid>1005</eventInfo.oid> <!-- CBPD -->
  <eventInfo.eventHostname>sensor1.net.cmu.edu</eventInfo.eventHostname>
  <eventInfo.eventClass>5</eventInfo.eventClass> <!-- Environmental -->
  <eventInfo.warningLevel>7</eventInfo.warningLevel> <!-- Informational -->
  <dataPayload.payloadType>2</dataPayload.payloadType> <!-- Cooked -->
  <dataPayload.payload>
    <cbpd>
      <cbpd-1.0.0>
        <heatSensor>
          <coverageArea>0.2</coverageArea>
          <heatSensorSetPoint>0.111</heatSensorSetPoint>
          <heatSensorLowerRange>0.003</heatSensorLowerRange>
          <heatSensorUpperRange>0.280</heatSensorUpperRange>
          <heatSensorAccuracy>0.001</heatSensorAccuracy>
          <timeConstant>0.1</timeConstant>
        </heatSensor>
      </cbpd-1.0.0>
    </cbpd>
  </dataPayload.payload>
</cerFactory>
```