Here is a fragment of the WSDL describing the service:

```xml
<wsdl:service name="CEM" address="/CEM">
  <wsdl:port name="CEMService" binding="tns:CEMFaultsBinding">
    <wsdl:operation name="JobStatusGet">
      <wsdl:input message="tns:CEMFaultsRequest"/>
      <wsdl:output message="tns:CEMFaultsResponse"/>
    </wsdl:operation>
  </wsdl:port>
</wsdl:service>
```

The WSDL describes the service operations for obtaining job status information from the CEM service. The `JobStatusGet` operation allows you to retrieve the status of a job by providing a job ID as input and receiving a response message containing the job status information.

**CREAM Services**

**Computing Resource Execution And Management service**

A lightweight service that implements all the operations at the Computing Element (CE) level; its well-defined WebService-based interface and its implementation as an extension of the Java-Axis servlet (running inside the Apache Tomcat container) provide interoperability with clients written in any programming language and running on any computer platform.

**CREAM API**

The CREAM interface is well-defined using the Web Service Description Language (WSDL); anyone can generate his/her CREAM client by simply filling in the stub code generated by the gSOAP tool (C++, Axis for Java, Perl module for perl).

Here is a fragment of the WSDL describing the service:

```xml
<wsoap:operation name="JobStatusGet">
  <wsdl:input message="tns:CEMFaultsRequest"/>
  <wsdl:output message="tns:CEMFaultsResponse"/>
</wsoap:operation>
```

**CREAM Architecture**

TrustManager extracts the user's credentials and other metadata from the user proxy certificate and forwards them to CREAM.

**CREAM Sequence Diagram**

The ICE layer subscribes to the CEMon service in order to asynchronously receive notifications about job status changes.

**WMS - CREAM Integration**

Thanks to ICE (a gSOAP/C++ intermediate layer) CREAM can receive job operations directly from a Grid WMS (Resource Broker). The ICE layer subscribes to the CEMon service in order to asynchronously receive notifications about job status changes.

In case some notifications are lost, ICE performs synchronous status polling for jobs for which it hasn't received status for some time. To maintain its subscriptions ICE periodically checks them and renews the expiring ones.

The status notification is sent by the component named CEMon[for]. It is a general purpose notification framework working in synchronous and asynchronous mode, that virtually supports any kind of monitoring thanks to its plug-in architecture.

Everyone can develop a sensor using CEMon's APIs, to be hot-plugged into the plug-in architecture. CEMon gets informations about job status from the CREAM data persistence backend through INDI APIs.

All communication with CREAM/CEMon in any direction is authenticated using SSL/TLS technology modified to support the grid proxies and implemented in the TrustManager running in the Tomcat container.

**CREAM Web Site:**

http://grid.pd.infn.it/cream/field.php

---

**Primary Authors:**

ANDRETTI, Paolo (INFN Padova) (andretti@pd.infn.it), BORGIA, Antonino Stefano (borgia@pd.infn.it), DORIGO, Alesio (dorigoa@pd.infn.it)

GIANELLE, Alessio (gianelle@pd.infn.it), MARASCHINI, Alessandro (maraschini@pd.infn.it)

MARZOLLA, Moreno (marzolla@pd.infn.it), MULAC, Michael (mulac@pd.infn.it)

PARRINI, Andrea (parrini@pd.infn.it)

---

**CREAM: A simple, Grid-accessible, Job Management System for local Computational Resources**

CHEP 2006, 13-17 February 2006, T.I.F.R. Mumbai, India