ECSA – Elearning Community Service Architecture

Heiko Bernlöhr

April, 12, 2011

Contents

1 Overview **2**

1.1 Sample usage scenario 3

2 Participants **5**

2.1 Basic functionalities and requirements 5

2.1.1 Technology / Architecture 5

2.1.2 Authentication 5

2.1.3 Authorization 6

2.1.4 ECS REST interface 6

2.1.5 Ressource extensions / Alterations 6

2.1.6 Web interfaces 6

2.2 Communication procedures / scenarios 6

2.2.1 Retrieving courselink information 7

Chapter 1

Overview

An ECSA is a service architecture for elearning based webservices. It provides mechanisms for communication and authorization between elearning systems among each other and management systems. This is implemented via a MOM.

The ECSA is derived from an architecture style for distributed systems called REST.

It will put major efforts that only recognized Web standards/protocols and web components are used, whereby a high degree of compatibility and connectivity is achieved. See figure 1.1 for ECSA components.

An ECSA builds up of three primary components:

- The ECS (elearning community server) serves the core functionionality of an ECSA network. It provides named message resources to allow communication between all participants.
- An ECC (elearning community client) is a participant in an ECSA network. It has to be registered at ECS and must be able to talk to the ECS as a REST based client. This participant normally has a native implementation of the ECS interface. Our favourite ECCs are LMSs (learning management systems).
- An ECP (elearning community proxy) represents a special kind of participant. It serves as a proxy for a none ECSA compliant system so

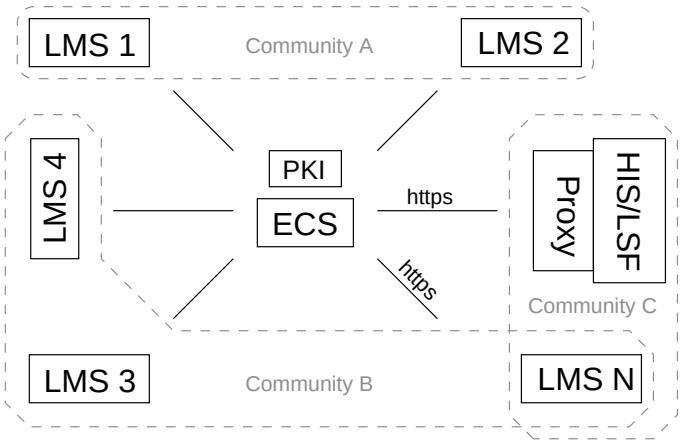

Figure 1.1: Components of an ECSA network.

that such a system is able to participate in an ECSA network without ever knowing about it.

1.1 Sample usage scenario

Suppose you have several LMSs (learning management systems) and want to share courses between them. You decide not to interchange the real courses but only course links which consist of some meta data of the appropriate course especially a link formed by an URL pointing to the real course so you can call it through the WWW e.g.:

`http://ilias.freeit.de/goto.php?target=pg_26_43&client_id=ecs2}`

Now it's possible for each LMS to communicate the released courses by the resources provided from the ECS to an explicit LMS (point to point) or to a community of LMSs (point to multipoint).

Because of the uniform application interface – there are only GET, PUT, DELETE and POST operations – receiving participants can fetch messages through a GET on the resource URL or sending messages by a POST on the resource URL (with some additional query parameters or header variables to point to the appropriate receivers).

To illustrate this we use the simple ECC application *curl* to send a message from one participant to another:

```
curl -i -H 'X-EcsAuthId: pid01' \
        -H 'X-EcsReceiverMemberships: mid02' \
        -H 'Content-Type: application/json' \
        -X POST \
        -d '{
              "name": "Mathematics II",
              "url" : "http://ilias...?target=pg_26_43&client_id=ecs2",
              ...
            }' \
        http://ecs.freeit.de/campusconnect/courselinks
```

In order to receive a message (in fifo mode) the receiving participant may call:

```
curl -i -H 'X-EcsAuthId: pid02' \
        -H 'Accept: text/plain; application/json' \
        -X GET \
        http://ecs.freeit.de/campusconnect/courselinks/fifo
```

Of course, there are several ways to operate on a resource. For details on using the resources located on an ECS and the different parameters (http headers, query strings) please see XXX for details.

Chapter 2

Participants

A particpant represents a legal client in an ECSA network.

2.1 Basic functionalities and requirements

2.1.1 Technology / Architecture

- has to communicate with the ECS as a REST client.
- HTTP 1.1 as transport and application protocol
- provide persistent connection (keep-alive)
- provide SSL/TLS transport layer
- has to use UTF-8 data encoding

2.1.2 Authentication

- HTTP Basic auth
- X.509 Certificates (SSL/TLS client authentication)

2.1.3 Authorization

A client should be able to use a simple "one touch token" authorization through the ECS /authtokens ressource. This authorization should be used either in redirecting users clicking on course links or maybe used in accessing participants in interconnected ECS networks.

2.1.4 ECS REST interface

2.1.5 Ressource extensions / Alterations

To make resource extensions and alteration possible the clients have to easily permit

- additional ressources
- extensible data formats
- Postels's Law (robustness principle):

  > Be conservative in what you send; be liberal in what you accept.

- versioning through request and response header (content negotiation)
  - Accept: application/vnd.my-format.v1+json
  - Accept: application/vnd.my-format.v2+json

2.1.6 Web interfaces

- Interface for ECS configuration data

2.2 Communication procedures / scenarios

In order to take part in an ECSA network a participant has to communicate with the ECS and other participants in different ways.

2.2.1 Retrieving courselink information

Figure 2.1 on page 8 shows the communication procedure how a LMS retrieve a courselink:

1 First the LMS fetches (POST) an event message from its event resource (`/events/fifo`) of ECS , which gives it a new or updated course resource meta data URL on ECS. Supposing this would be `/campusconnect/course/5`.

1.1 Now the LMS takes this URL and fetch (GET) it from ECS (the LMS only fetch the message via a GET, so that the message will still be there). Only now the LMS get the real resource URL to fetch the desired course data from the proxy. This url maybe itself an encoded url: `https://.../58680c636c8bc4a16e047d758f2e7773118fa141`

1.2 Next the LMS fetches (POST) a one touch token from the `/auths` resource of ECS in case the proxy use it for authorization against ECS.

1.3 Then the LMS get (GET) the actual course data from the proxy URL provided by the received message in 1.1 .

1.4 Until it will get back the course resource representation in 1.3 successfully, it deletes (DELETE) the message `/campusconnect/course/5` received in 1.1 on ECS.

This procedure guranties that the appropriate course data will remain on the proxy until the LMS has successfully fetched the data, because after the message `/campusconnect/course/5` has been deleted from the LMS the proxy will be informed from the ECS, so that the real course data could be deleted. Of course this information occurs only if all addressed participants has successfully fetched the message on the proxy and if the `/campusconnect/course` resource is not tagged as a postrouted resource.

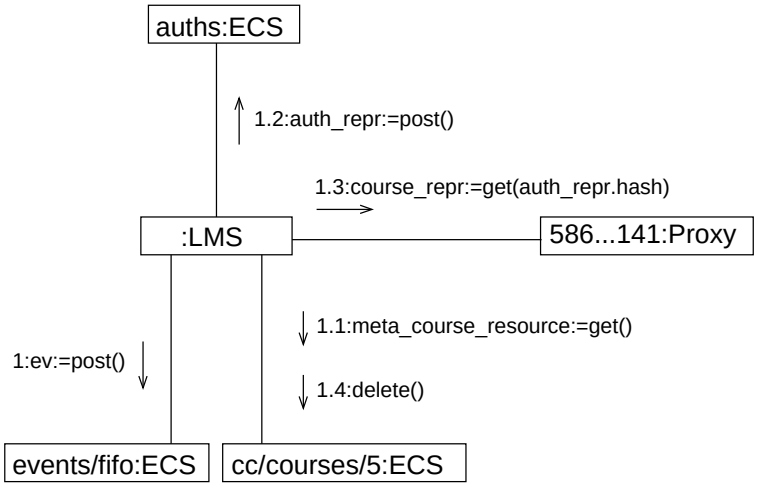

Figure 2.1: LMS courselink retrieval communication procedure.

List of Figures

1.1 ECSA components. 3

2.1 LMS courselink retrieval 8

List of Symbols

ECC elearning community client

ECP elearning community proxy

ECS elearning community server

ECSA elearning community service architecture

LMS learning management system

MOM message orientated middleware

participant a client in ECSA network

Index

ECSA
    overview, 2

participant, 5
    architecture, 5
    authentication, 5
    authorization, 6
    communication, 6
    ressource extensions, 6
    retrieving courselink, 7
    technology, 5