Laverca

Tekninen kuvaus

Eemeli Miettinen (eemeli@methics.fi)
July 23, 2013

Sisällys

Muutoshistoria

Versio	Pvm	Tekijä	Huom
0.9	7.4.2011	Asko Saura, Methics Oy	Ensimmäinen kokonainen Laverca-versio. Valmis operaattorien tarkastettavaksi.
1.0	28.6.2011	Asko Saura, Methics Oy	Julkaisuvalmis versio.
1.2	23.7.2013	Eemeli Miettinen	Päivitys vastaamaan paremmin nykytilannetta.

1 Johdanto

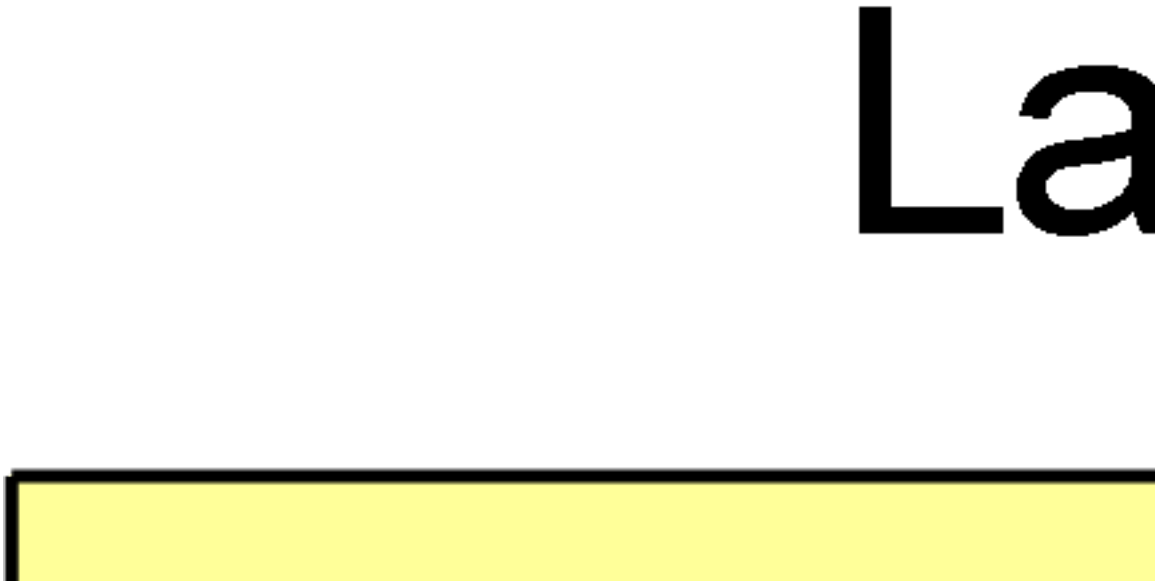
Laverca on Open Source –toteutus ETSI TS 102 204-client –ohjelmistoksi Javalla. Vakio-ETSIn lisäksi se sisältää FiCom 2.0 –soveltamisohjeen toiminnallisuuden ja pyrkii tarjoamaan FiComin tavallisimmat asiat suoraviivaisina kehittäjille, erityisesti siten, ettei kehittäjien tarvitsisi edes tietää C/S Async messagingmoden käyttöperiaatetta ja siten, että kehittäjät suoraan näkevät FiComin tarjoamat palvelut Laverca-kirjaston muodosta. Tässä paperissa kuvataan Lavercan tekninen toteutus.

Tässä paperissa puhutaan yksinomaan Javalla toteutettavasta kirjastosta.

Tämä paperi on kirjoitettu suomeksi, koska FiCom –kirjallisuus oli kirjoitushetkellä saatavilla vain suomeksi.

2 Arkkitehtuuri

Lavercan arkkitehtuuri peilaa suoraan sitä kirjallisuutta, johon ohjelmisto nojautuu. Suurin painopiste on FiCom-soveltamisohjeella, jonka toiminnallisuutta varten Lavercan arkkitehtuurissa on oma erityinen kerroksensa.



Kuva 1. Lavercan kerrosarkkitehtuuri suhteessa kirjallisuuteen.

Kerroksittainen arkkitehtuuri on valittu todennäköisiä asennuskohtaisia räätälöintejä ja tulevaisuuden kehitystä silmälläpitäen. Laverca tukee FiCom –toiminnallisuuksia hyvin, mutta sallii kehittäjien käyttää niiden allaolevaa ETSI-kerrosta suoraankin.

Alakerrokset ovat varsin suoraviivaisia; niissä toteutus pyrkii ensisijaisesti noudattamaan standardeja. FiCom-kerroksessa on Lavercan painopiste; siellä luomme Laverca –projektin varsinaisen lisäarvon ja siellä asioita voisi tehdä monellakin eri tavalla. Perusratkaisuna Lavercassa on C/S-viestintätavan piilottaminen Javan vakiovälineiden (java.util.concurrent) avulla.

3 FiCom -kerros

Lähimpänä Laverca –kirjaston käyttäjää on FiCom API eli ns. FiCom –kerros. Se tarjoaa vain muutamia funktioita, jotka vastaavat suoraan FiCom –soveltamisohjeen tarjoamia viittä signature profilea.

authenticate(..)
authenticateAnon(..)
signText(..)
signData(..)
consent(..)

Kerroksen ideana on tarjota sovelluskehittäjille suoraviivaisesti käytettävät korkean tason funktiot, jotka piilottavat alleen ETSIn yksityiskohdat, tärkeimpänä C/S async –messagingmoden. Käyttäjä näkee kerroksen tarjoaman funktion asynkronisena metodikutsuna.

```
import fi.laverca.*;
import org.etsi.uri.TS102204.v1_1_2.Service;
...

String apId  = "http://my_ap_id";
String apPwd = "my_ap_pass";
String sigUrl = "https://www.myae.fi/soap/services/MSS_SignaturePort";
String statUrl= "https://www.myae.fi/soap/services/MSS_StatusQueryPort";
String statUrl= "https://www.myae.fi/soap/services/MSS_ReceiptPort";

FiComClient fiComClient = new FiComClient(apId, apPwd, sigUrl, statUrl, recUrl);
.. setup SSL

String phoneNumber = .. ask user for it ..
String nospamCode = .. ditto ..
String apTransId = .. generate unique id for this transaction ..
String challenge = .. generate authentication challenge ..
String eventId = .. generate unique id for this transaction ..

Service noSpamService = FiComAdditionalServices.createNoSpamService(.., false);
Service eventIdService = FiComAdditionalServices.createEventIdService(eventId);

try {
    FiComRequest req = fiComClient.authenticate(apTransId,
                                                challenge,
                                                phoneNumber,
                                                noSpamService,
                                                eventIdService,
                                                null, // additionalServices,
                                                new FiComResponseHandler() {
        @Override
        public void onResponse(FiComRequest req, FiComResponse resp) {
           log.debug("got resp");
           log.debug(resp.getPkcs7Signature().getSignerCn());
        }
        @Override
        public void onError(FiComRequest req, Throwable throwable) {
            log.debug("got error", throwable);
        }
        @Override
        public void onOutstandingProgress(FicomRequest req, ProgressUpdate prg) {
            log.debug(“got progress update”);
        }
    });
}
```

```
catch (IOException e) {
    log.debug("error establishing connection", e);
}
```

Toimiva esimerkki löytyy Laverca-jakelusta
~/examples/src/fi/laverca/examples/FiComSigReqCaller.java

4 ETSI-kerros

FiCom –kerroksen alla on Lavercan varsinainen tekninen sydän, joka toteuttaa ETSI TS 102 204 –tietotyypit sekä SOAP-kutsut. Täällä on koneisto, jonka avulla kerroksen käyttäjä (erityisesti FiCom –kerros) saa aikaan 204-liikennettä. Tämän koneiston suunnittelu ja toteutus ovat suoraviivaisia standardien noudattamisharjoituksia eikä täällä edes yritetä mitään sen erityisempää tai ”luovaa lisäarvon tuottamista”. Tämä on perustekniikkaa. Alla esimerkki siitä, miten tämän kerroksen käyttäjä toteuttaa synkronisen allekirjoitustapahtuman.

```
import fi.laverca.*;
...

String apId  = "http://my_ap_id";
String apPwd = "my_ap_pass";
String sigUrl = "https://www.myae.fi/soap/services/MSS_SignaturePort";
String statUrl= "https://www.myae.fi/soap/services/MSS_StatusQueryPort";
.. all the other URLs

EtsiClient etsiClient = new EtsiClient(apId,
                                       apPwd,
                                       aeMsspIdUri,
                                       msspSignatureUrl,
                                       msspStatusUrl,
                                       msspReceiptUrl,
                                       msspRegistrationUrl,
                                       msspProfileUrl,
                                       msspHandshakeUrl);

String apTransId = "A"+System.currentTimeMillis();
String msisdn = "+35847001001";
DTBS dtbs = new DTBS("sign this", "UTF-8");
String dataToBeDisplayed = null;
String signatureProfile = FiComSignatureProfiles.SIGNATURE;
String mss_format = FiComMSS_Formats.PKCS7;
MessagingModeType messagingMode = MessagingModeType.SYNCH;

MSS_SignatureReq sigReq =
  etsiClient.createSignatureRequest(apTransId,
                                    msisdn,
                                    dtbs,
                                    dataToBeDisplayed,
                                    signatureProfile,
                                    mss_format,
                                    messagingMode);

MSS_SignatureResp sigResp = null;
try {
    sigResp = etsiClient.send(sigReq);
}
catch(AxisFault af) {
    log.error("got soap fault", af);
    return;
}
catch(IOException ioe) {
    log.error("got IOException ", ioe);
    return;
```

```
}
```

Toimiva esimerkki löytyy Laverca-jakelusta
~/examples/src/fi/laverca/examples/EtsiSigReqCaller.java

Muut kerrokset

SOAP-toteutus perustuu Apache Axis 1.4 ja Castor 1.3 –kirjastoihin. Axis huolehtii SOAP-liikenteestä ja Castor tekee Java-XML-Java –muunnokset tiedolle.

Axis käyttää HTTP-yhteyksissään Apache HttpClient –kirjastoa Javan vakiotyökalujen asemesta. Laverca –projektissa käytetään muutamaa muutakin Axiksen vaatimaa työkalukirjastoa.

SSL/TLS –toteutuksessa Laverca nojaa Javan vakio-SSL-tukeen eli käytännössä ympäristömuuttujilla säädettävään omaan SSL-identiteettiin ja luottaa Truststore –tiedoston avulla operaattorien AE-palvelujen SSL-varmenteisiin. SSL-tuen käyttöä varten on oma luokkansa fi.laverca.JvmSsl .

5 Paketointi

Laverca –jakelu sisältää seuraavat osat

- laverca-core.jar
  laverca-datatypes.jar
- libs -hakemiston kirjastot
- esimerkkisovellus
- apidocit
- tämä dokumentti
- MSS FiCom Soveltamisohje v2.0

Paketti sisältää Methics Oy:n ylläpitämän demopalvelimen AE-yhteystiedot.

6 Tekniikan vaikutukset lisensointiin

Laverca käyttää hyväkseen olemassaolevia open source –kirjastoja. Niiden lisenssiehdot asettavat rajoja sille, miten Lavercaa voidaan lisensoida ja jaella.

Tässä läpikäydään merkittävimpien kirjastojen lisensointi. Muitakin kirjastoja on, ja täydelliset lisenssitiedot ovat docs/licenses/ -hakemistossa.

Kirjasto	Lisenssi	Huom
Apache Axis	©Apache Software Foundation Apache License, Version 2.0	axis-src-1_4.tar.gz/LICENSE
Castor	©Werner Guttman Apache License, Version 2.0	http://www.castor.org/license.html
Apache HttpClient	©Apache Software Foundation Apache License, Version 2.0	http://hc.apache.org/