Trustedhow Location Based Services (“Location Aware Signature”)

Prototype B: NFC-enabled Crypto Tag

christian.Lesjak@student.tugraz.at

January 2013

Development Environment

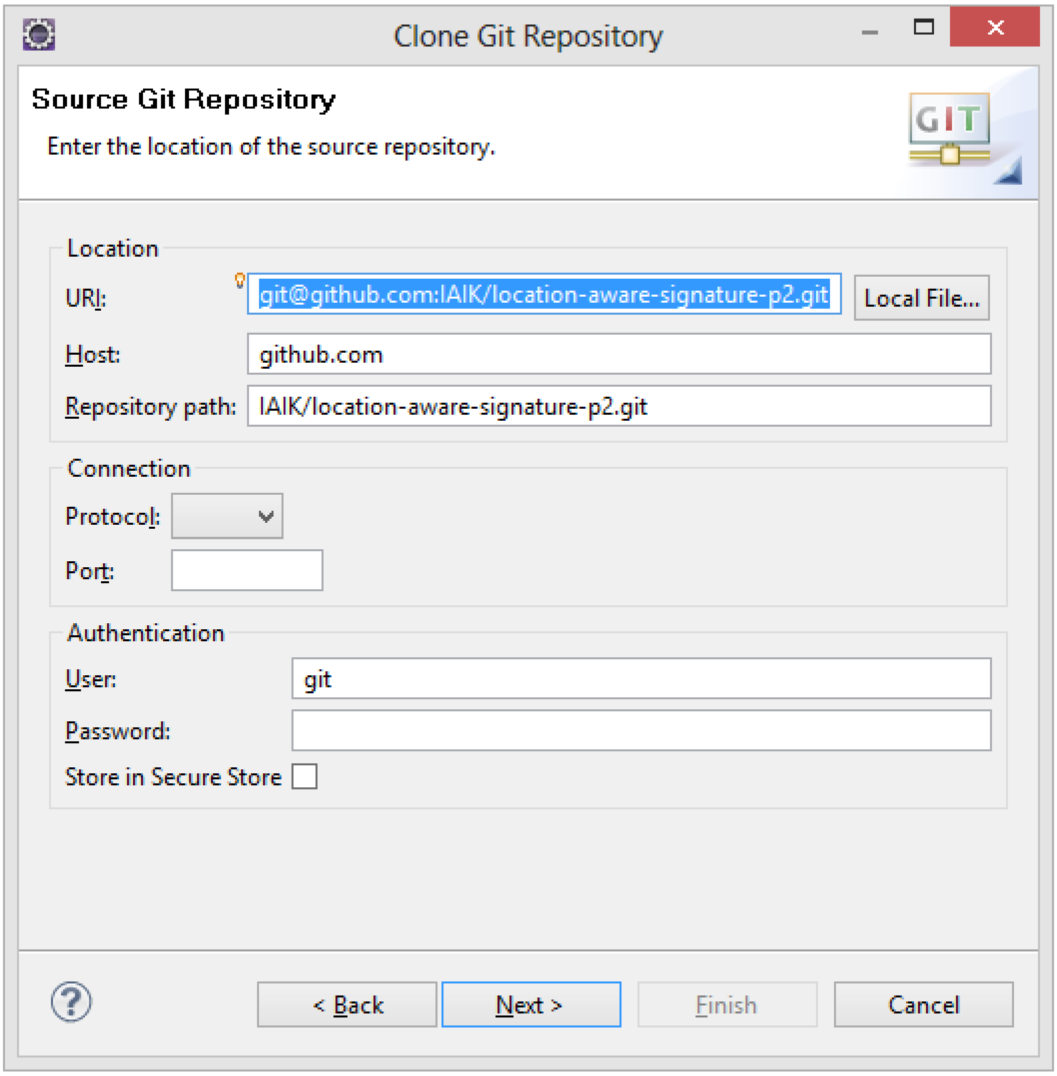
Software

- JDK 7
- Eclipse Java EE IDE for Web Developers, Indigo Service Release 2
- Android SDK, Revision 21
- Android Development Tools (ADT) for Eclipse, Revision 21
- Google App Engine SDK, Version 1.7

Check-out

In Eclipse, open the “Git Repositories” view and select “Clone a Git repository”. In the upcoming dialog, select “URI” as source and then supply the following options:

URI: git@github.com:IAIK/trusted-location-based-services.git



Source Git Repository

Enter the location of the source repository.

Location

URI: git@github.com:IAIK/location-aware-signature-p2.git

Host: github.com

Repository path: IAIK/location-aware-signature-p2.git

Connection

Protocol:

Port:

Authentication

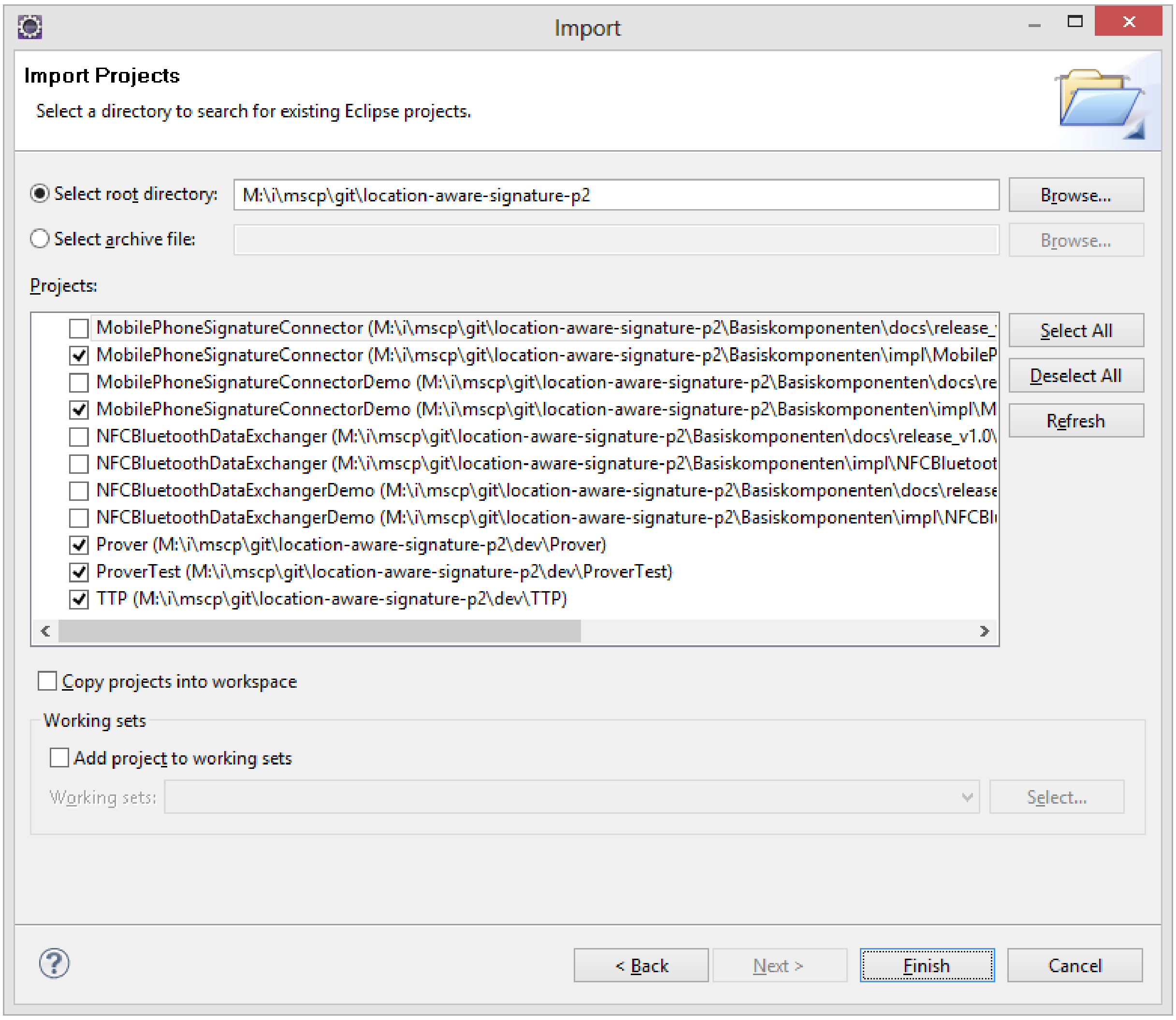
User: git

Password:

Store in Secure Store □

On the final dialog (“Local Destination”), choose a local folder for the repository and proceed with default options.

To import the projects into your Eclipse workspace, use the Import function with “Existing Projects into Workspace” and select your local git respository as root directory:



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory: M:\i\mscp\git\location-aware-signature-p2

○ Select archive file:

Projects:

- □ MobilePhoneSignatureConnector (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\docs\release_
- ☑ MobilePhoneSignatureConnector (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\impl\MobileP
- □ MobilePhoneSignatureConnectorDemo (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\docs\re
- ☑ MobilePhoneSignatureConnectorDemo (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\impl\M
- □ NFCBluetoothDataExchanger (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\docs\release_v1.0\
- □ NFCBluetoothDataExchanger (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\impl\NFCBluetoot
- □ NFCBluetoothDataExchangerDemo (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\docs\release
- □ NFCBluetoothDataExchangerDemo (M:\i\mscp\git\location-aware-signature-p2\Basiskomponenten\impl\NFCBl
- ☑ Prover (M:\i\mscp\git\location-aware-signature-p2\dev\Prover)
- ☑ ProverTest (M:\i\mscp\git\location-aware-signature-p2\dev\ProverTest)
- ☑ TTP (M:\i\mscp\git\location-aware-signature-p2\dev\TTP)

□ Copy projects into workspace

Working sets

□ Add project to working sets

Working sets:

Project Structure

- **Basiskomponenten**
  Library to deal with Austrian Mobile Phone Signature.
- **dev**
  - **Prover**: Android Client that communicates with the Crypto Tag
  - **ProverTest**: Android Test project with unit tests for the Prover app.
  - **TTP**: Server that runs the protocol and stores a database of all crypto tags. Written in GWT, accessible via web services (protocol) and web (user interface). In this prototype, the server also incorporates a simple service provider (log of issued T-LTTs).
- **doc**
  This documentation, screenshots, mock tags and the presentation.

Create a Google Account for your smartphone

Design

Terms

Crypto tag: Referring to any kind of NFC-enabled smart card that is capable of cryptographic primitives.

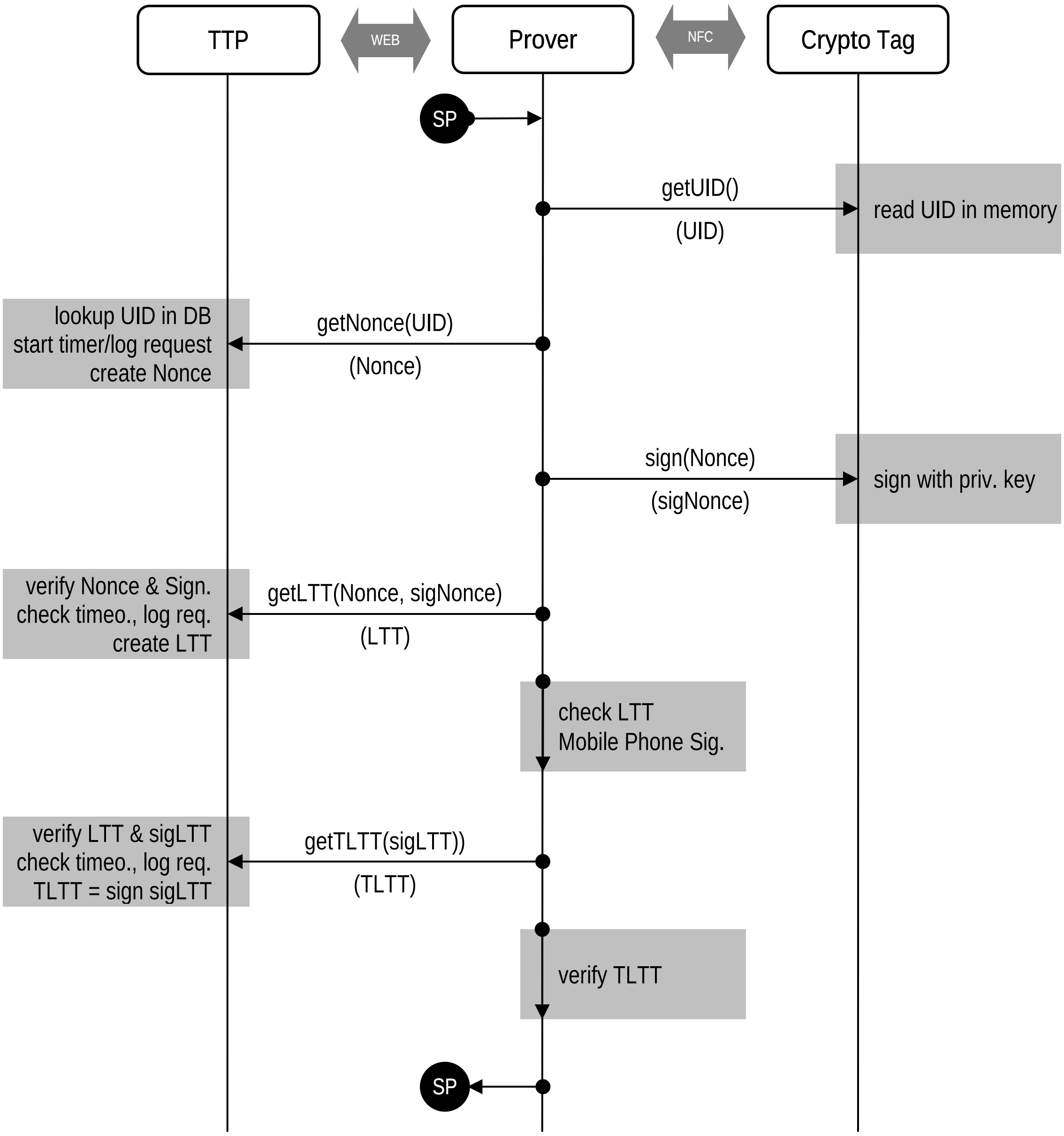
Crypta: A specific crypto tag (cryptographic protected tag) developed by the IAIK and austriamicrosystems.

Prover: Smartphone application and it's user who wants to prove his location.

TTP: Trusted Third Party, in this prototype composed of the crypto tag and the backend-server running on GAE.

For details about Trusted Location Based services, see paper by Peter Teufel et al.

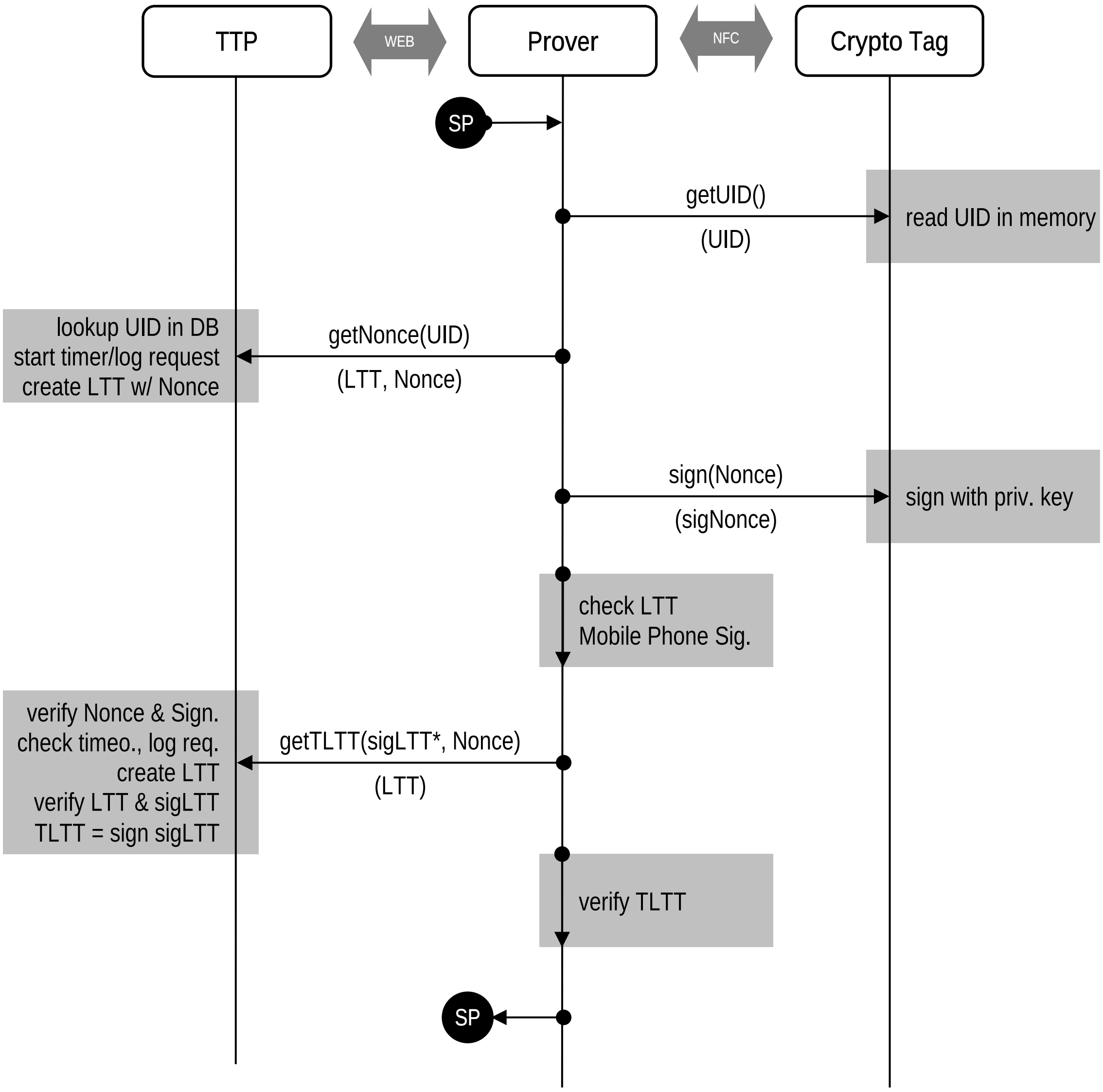
Protocol



Improved Version of Protocol (not implemented)

Before Prover requests T-LTT by sending getTLtt(signedLTT) to the TTP Server, an attacker might send an unmodified LTT, but signed by himself, to the TTP.

To fix this issue, the signedNonce could be integrated into the signedLtt* by the Prover before applying the Mobile Phone Signature.



Implementation

Libraries

RPC-Communication between Android and GAE (TTP API)

- Hessian (on GAE): http://hessian.caucho.com/
- Hessdroid (on Prover smartphone app): http://code.google.com/p/hessdroid/

Cryptographic operations:

- IAIK JCE and IAIK ECC (on GAE and in Prover app)

Utilites

- Apache Commons FileUpload (http://commons.apache.org/fileupload/) on GAE for certificate upload.
- SimpleXML (http://simple.sourceforge.net/) on GAE and in Prover for XML parsing as Android provides no JAXB.
- appengine-util (http://www.resmarksystems.com/code/) to allow Text fields to be serialized (to transfer them from GAE server to GWT client side).

Deployment on GAE

Web UI (TTP Server)

To deploy the project to GAE, right click the TTP project in Eclipse, choose “Google” > “Deploy to App Engine”. You first need to add your Google account (the Nexus S Google Account from above does not work – the account must have been verified with a phone number first). For testing, I used my personal account (which has an associated phone number).

Test Data

Before running the protocol, make some either real or mock tags have been added on the administrative web interface of the TTP.

Some sample locations:

- Semriach (latitude = 47…, longitude = 15…)
  https://maps.google.com/?ll=**47**.219936,**15**.401773
- IAIK
  https://maps.google.com/?ll=**47**.058421,**15**.457815

IAIK Crypta Tag UID: 3F0800A0A1A2A3

Use Cases

Generate Fake Tags (“Mock Tags”)

In the Provera app, choose „Create Mock Tag” from the menu. The generated tag will be store on the external storage of the smart phone (sdcard/LAS-Prover/mocktags/<UID>/).

Add tags to the TTP

To run the protocol, add either real crypto tags or mock tags to the TTP. On the web interface go to Management/Add Tag. Fill out the data accordingly, and upload the tag’s certificate (for generated mock tags from the phone’s external storage in sdcard/LAS-Prover/mocktags/<UID>/PublicKeyCert.cer.

Acquire an T-LTT

To initiate the protocol to acquire a Trusted Location Time Ticket (T-LTT), open the Prover app and either touch a real crypto tag or select a mock tag from the menu.

The acquired T-LTT can be viewed on the web interface of the TTP under History/Issued T-LTTs.

Common Issues

Various helpful resources for re-occurring issues during GWT/GAE/Android development.

Conversion to Dalvik format failed

Problem:

[2012-10-02 11:16:53 - Prover] Conversion to Dalvik format failed with error 1

Solution:

http://stackoverflow.com/questions/2680827/conversion-to-dalvik-format-failed-with-error-1-on-external-jar

In my case this specific answer was helpful: http://stackoverflow.com/a/8106366

NoClassDefFoundError

Problem:

NoClassDefFoundError with classes from Hessian

Solution:

http://stackoverflow.com/a/9916751

JDO and how to serialize these objects for GWT

Problem: Serialization/RPC/Data objects

Solution: https://developers.google.com/web-toolkit/doc/1.6/DevGuideServerCommunication

https://developers.google.com/appengine/docs/java/datastore/jdo/dataclasses

No source code is available for type xxx

Problem: No source code is available for type xxx did you forget to inherit a required module?

Solution: move data objects to client package

Sync datastore between test environment (hosted mode) and GAE live system

Problem: Sync datastore between test environment (hosted mode) and GAE live system

Solution: Probably not possible so easy

As has been discussed in several forums, GWT does not emulate any AppEngine classes, so there are a few possible workarounds:

1. Create a DTO that does not contain any AppEngine classes for each domain object. DTOs, yuck.

2. Provide an emulated Key class yourself following Fred Sauer’s fine instructions.

3. Use the GILEAD project’s Adapter4AppEngine, which supplies GWT with emulated classes for AppEngine’s Key, Text, Blob, and other troublesome types.

GWT & JDO key string

Solution:

Sriran Narayan says to String-encode the Key to get it to pass through GWT's RPC mechanism:

http://stackoverflow.com/questions/988217/gwt-with-jdo-problem

GWT Examples

http://gwt.google.com/samples/Showcase

GAE Object Manager has been closed

Google App Engine error Object Manager has been closed

http://stackoverflow.com/questions/2968868/google-app-engine-error-object-manager-has-been-closed

File Upload to GWT/GAE

http://commons.apache.org/fileupload/

http://commons.apache.org/io/

Server response on file upload is:

<pre style="word-wrap: break-word; white-space: pre-wrap;">agdsYXMtdHRwchcLEhFUYWdDZXJ0aWZpY2F0ZUpkbxgaDA</pre>

Solution:

Manually parsing it out using two custom tags for start end end of my actual payload

javax.jdo.JDOFatalUserException

javax.jdo.JDOFatalUserException: Detected attempt to establish TagJdo(34) as the parent of TagCertificateJdo(33) but the entity identified by TagCertificateJdo(33) has already been persisted without a parent. A parent cannot be established or changed once an object has been persisted.

NestedThrowables:

org.datanucleus.store.appengine.DatastoreRelationFieldManager$ChildWithoutParentException: Detected attempt to establish TagJdo(34) as the parent of TagCertificateJdo(33) but the entity identified by TagCertificateJdo(33) has already been persisted without a parent. A parent cannot be established or changed once an object has been persisted.

Solution:

correctly annotate jdo objects: https://developers.google.com/appengine/docs/java/datastore/jdo/relationships

java.security.AccessControlException with IAIK JCE

java.security.AccessControlException: access denied on Security.addProvider(new iaik.security.provider.IAIK());

Solution:

final solution: use GAE SDK 1.7.2 and disable precompilation in appengine-web.xml!

http://code.google.com/p/googleappengine/issues/detail?id=1612

https://developers.google.com/appengine/docs/java/runtime#no_signed_jars http://code.google.com/p/googleappengine/issues/detail?id=2889

Since the unlimited strength jurisdicition policy files for JCE are not installed, crypto algorithms are usable, but limited in strength (example:

maximum 128 bits keys for AES).

It would be nice if the unlimited strength jurisdicition policy files were installed on the production servers. They can be installed in the local Java SDK by the users themselves.

http://grokbase.com/t/gg/google-appengine/126ttq24tf/is-strong-symmetric-encryption-possible-on-gae-at-all-using-java-ive-searched-for-hours-now

SDK 1.7 enables you to register your own JCE provider, so it now works with Bouncy Castle for instance. Just make sure to disable precompilation, since it breaks signed JARs.

https://groups.google.com/forum/?fromgroups=#!topic/google-appengine-java/CYZRGgml78w

https://groups.google.com/forum/?fromgroups=#!msg/google-appengine-java/CYZRGgml78w/tpWWHn9KkwIJ

http://dmitrygusev.blogspot.co.at/2009/08/turn-java-security-manager-off-in.html

http://stackoverflow.com/questions/1454519/is-there-a-right-way-to-manipulate-googleappengine-security-permissions

AsyncTask and Unit Tests

Android AsyncTask testing problem with Android Test Framework

http://stackoverflow.com/a/3802487/1730765

DataNucleus data enhancer

Persistent class "Class at.tugraz.iaik.las.p2.ttp.client.data.ProtocolLogJdo does not seem to have been enhanced. You may want to rerun the enhancer and check for errors in the output." has no table in the database, but the operation requires it. Please check the specification of the MetaData for this class.

```
java.lang.RuntimeException: Unexpected exception
        at com.google.appengine.tools.enhancer.Enhancer.execute(Enhancer.java:76)
        at com.google.appengine.tools.enhancer.Enhance.<init>(Enhance.java:71)
        at com.google.appengine.tools.enhancer.Enhance.main(Enhance.java:51)
Caused by: java.lang.reflect.InvocationTargetException
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source)
        at java.lang.reflect.Method.invoke(Unknown Source)
        at com.google.appengine.tools.enhancer.Enhancer.execute(Enhancer.java:74)
        ... 2 more
Caused by: java.lang.NoSuchFieldError: NUCLEUS_CONTEXT_LOADER
        at org.datanucleus.NucleusContext.<clinit>(NucleusContext.java:75)
        at org.datanucleus.enhancer.DataNucleusEnhancer.<init>(DataNucleusEnhancer.java:160)
        at org.datanucleus.enhancer.DataNucleusEnhancer.main(DataNucleusEnhancer.java:1133)
        ... 7 more
```

http://stackoverflow.com/questions/11286020/datanucleus-using-wrong-enhancer-in-google-app-engine-1-7

JDO warnings

The datastore does not support joins and therefore cannot honor requests to place related objects in the default fetch group. The field will be fetched lazily on first access.

I need to live with that warning, everything works fine anyhow.

http://stackoverflow.com/questions/9049491/org-datanucleus-exceptions-nucleususerexception-object-manager-has-been-closed

Conversion to Dalvik format failed

Conversion to Dalvik format failed: Unable to execute dex: Java heap space

http://stackoverflow.com/questions/5943712/conversion-to-dalvik-format-failed-unable-to-execute-dex-java-heap-space

Opening sockets on GAE

Permission denied: Not allowed to issue a socket connect locally

http://code.google.com/p/googleappengine/issues/detail?id=8166

2. Add the following filter config in the web.xml. ***You will need to remove*** this filter before uploading because this class is only available in the development environment.

```
  <filter>
    <filter-name>_ah_DevSocketFilter</filter-name>
    <filter-class>
      com.google.appengine.api.socket.dev.DevSocketFilter
    </filter-class>
    <init-param>
      <param-name>use-native-sockets</param-name>
      <param-value>true</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>_ah_DevSocketFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
```

Modelling Accuracy

http://gis.stackexchange.com/questions/8650/how-to-measure-the-accuracy-of-latitude-and-longitude

decimal places	degrees	distance
0	1	111 km
1	0.1	11.1 km
2	0.01	1.11 km
3	0.001	111 m
4	0.0001	11.1 m
5	0.00001	1.11 m
6	0.000001	0.111 m
7	0.0000001	1.11 cm
8	0.00000001	1.11 mm

GC overhead limit exceeded

Unable to execute dex: GC overhead limit exceeded

http://stackoverflow.com/questions/9471194/unable-to-execute-dex-gc-overhead-limit-exceeded

Serializeable GAE Text field

Fortunately a guy has already implemented such a serializable version of Text and provides it for free. I just tried it out and it works seamlessly, without writing a single additional line of code. Here some steps on how to use it (available descriptions on the web are really bad).

Download the necessary jar files from http://www.resmarksystems.com/code/:

appengine-utils-client-1.0.jar

appengine-utils-server-1.0.jar

Include the appengine-utils-client-1.0.jar in your build path. Copy the appengine-utils-server-1.0.jar to your WEB-INF/lib folder.

On your GWT module file add the following:

```
<inherits name="com.resmarksystems.AppEngineDataTypes"/>
```

Restart your GWT app or compile it and everything should work as expected.

http://juristr.com/blog/2010/02/gwt-app-engine-and-app-engine-data/

http://www.resmarksystems.com/code/

Various related Books

- Cryptography engineering : design principles and practical applications (Ferguson, Niels)
- Understanding cryptography : a textbook for students and practitioners (Paar, Christof)
- Beginning Cryptography with Java (Programmer to Programmer) [Taschenbuch] David Hook
- Cryptography: Theory and Practice (Discrete Mathematics and Its Applications) [Gebundene Ausgabe] Douglas Stinson
- Modern Cryptography: Theory and Practice (Hewlett-Packard Professional Books) [Gebundene Ausgabe] Wenbo Mao
- Beginning Cryptography with Java (Programmer to Programmer) [Taschenbuch] David Hook
- Practical Cryptography [Taschenbuch] Niels Ferguson
- Applied cryptography : protocols, algorithms, and source code in C (Bruce Schneier)
- Secrets and lies : digital security in a networked world; [with new information about post-9/11 security] (Bruce Schneier)