INTEL OpenSource TECHNOLOGY CENTER

# Single Sign-On Framework in Tizen

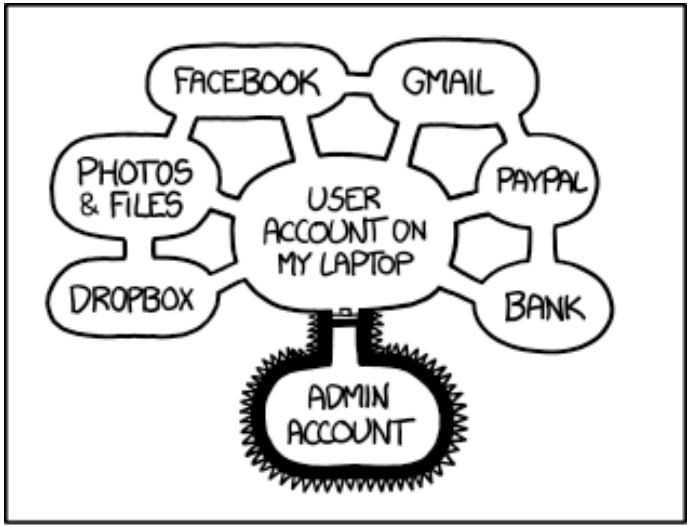
Contributors: Alexander Kanavin, Jussi Laako, Jaska Uimonen

Introduction

Architecture

Demonstration

# What is the problem that Single Sign-on systems are aiming to solve?




Source: xkcd.com/1200

# What is gSSO?

- A system for storing sensitive user data (credentials is the term for it) securely*
- Provides implementations of common** operations and authentication protocols that use those credentials and the API to access it
- Written entirely in C using Glib and its class and object system (Gobject)
- License: LGPL 2.1+

* the word 'securely' will be explained in a moment

** what exactly is meant by 'common' operations will also be explained in a moment

# What is 'security' as understood by gSSO? (1/2)

Credentials are stored in a database, and the database must be stored on disk so that only gsso can access it

- The details of such secure storage are handled by plugins
- By default classic Unix permissions on the database file are used
- The database can also be encrypted on disk using ecryptfs
- Other platform security mechanisms can be utilized by writing additional plugins

# What is 'security' as understood by gSSO? (2/2)

Access to each credential must be allowed only to explicitly listed applications

- Each credential has an access control list
- Checks against this list are performed by an access control plugin
- The default plugin is using filesystem paths to perform checks
- There is also a plugin that is using SMACK labels
- There is also support for access controlling applications, if they are not standalone binaries, but scripts that run in a runtime

# What are the common operations that use credentials?

- Applications do not access credentials directly (again, this is good for security)
- Applications initiate an operation on a credential from a list of allowed operations (mechanisms)
- List of allowed methods/mechanisms is stored together with the credential in a database
- Method is a class implemented by a plugin
- Mechanisms are functions of a method

With that said, the methods fall broadly into two categories:

- Offline (do not send anything over the network)
- Online (usually, authentication protocols to get access to some service)

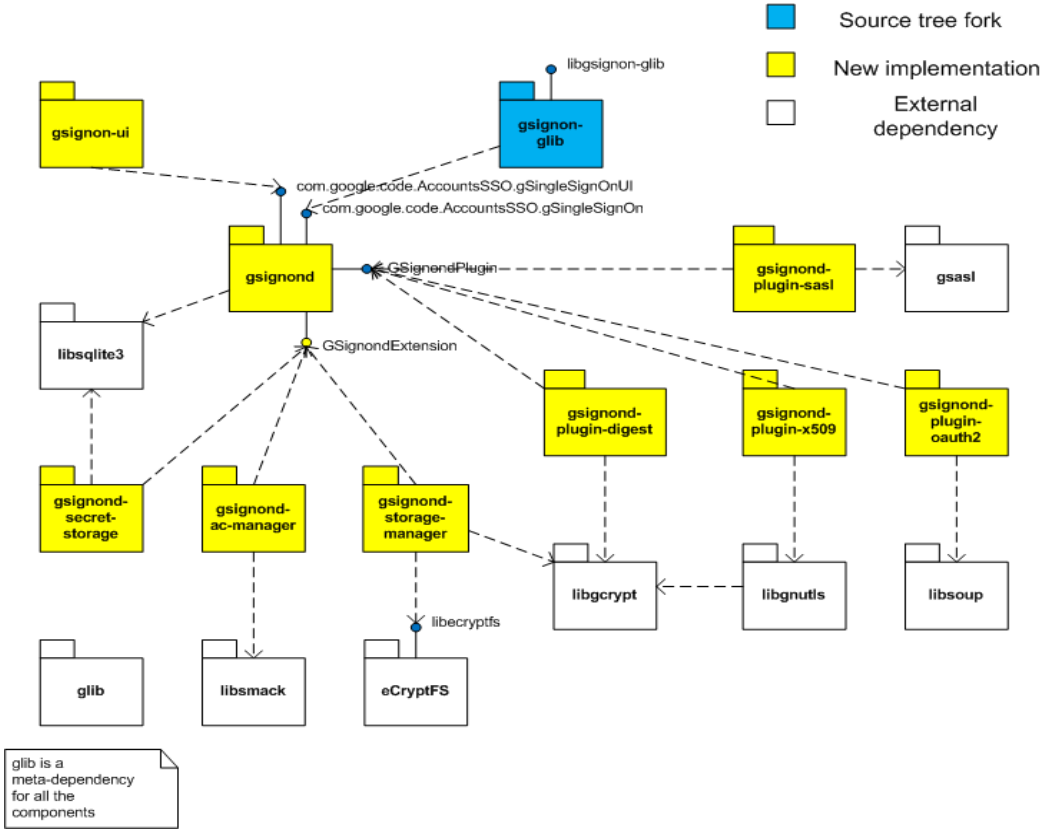
# Offline methods

- The simplest method: store and retrieve a username/password pair
  - Can be used for password management, or when the password is sent directly over the network (not a good idea!)
  - Implemented by 'password' plugin
- X.509
  - Handles operations with X.509 certificates such as sign, verify, encrypt and decrypt, without exposing the related keys to applications
- Generic encryption/decryption engine
  - The key is never exposed to the applications
- Front-end to specialized security hardware (such as a trusted execution engine or a smart card)
  - Provides a common API to applications, so they don't have to implement a hardware specific API

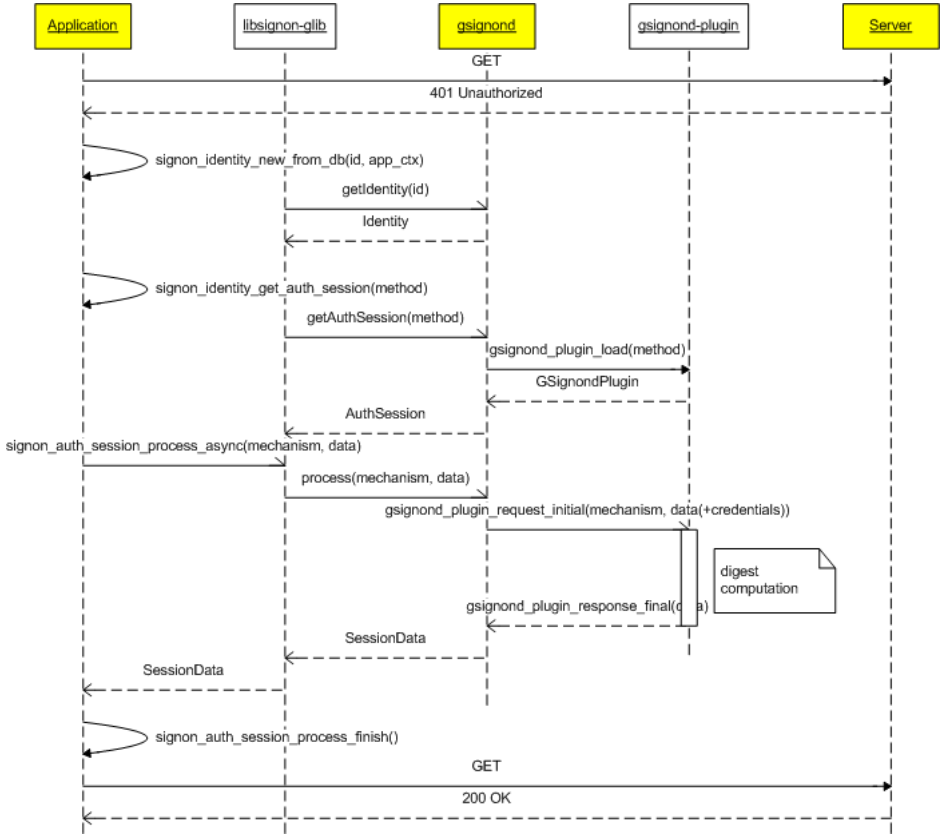
# Online methods

- OAuth version 1 and 2
  - Very popular mechanism for authorizing applications to access online services. Used by Google, Facebook, Twitter, Microsoft, LinkedIn, Amazon, Yahoo,... pretty much everybody.
  - Typically involves showing the user a webpage which asks if the user trusts some application to access some data from a service, and if the user does, the application gets a magic access string (a 'token')
  - GSSO implements the client side of OAuth 1 and 2 RFC standards fully, even though they are rather large :)
- SASL
  - A set of mechanisms for challenge/response based authentication
  - Used in IMAP, SMTP, XMPP, LDAP, IRC,...
  - GSSO implements the most common mechanisms
- HTTP Digest authentication

# Architecture – all the pieces together



# Architecture – functional flow



# Demo placeholder! (see notes)

# Conclusion

http://01.org/gsso

LGPL 2.1+

Git repositories, mailing lists, bug tracker, IRC channel, documentation, tarball downloads, etc.

All of that can be found at the above address