**CAS protocol**

The CAS protocol is a simple and powerful ticket-based protocol developed exclusively for CAS. A complete protocol specification may be found [here](http://jasig.github.io/cas/development/protocol/CAS-Protocol-Specification.html).

It involves one or many clients and one server. Clients are embedded in *CASified* applications (called “CAS services”) whereas the CAS server is a standalone component:

* The [CAS server](http://jasig.github.io/cas/development/installation/Configuring-Authentication-Components.html) is responsible for authenticating users and granting accesses to applications
* The [CAS clients](http://jasig.github.io/cas/development/integration/CAS-Clients.html) protect the CAS applications and retrieve the identity of the granted users from the CAS server.

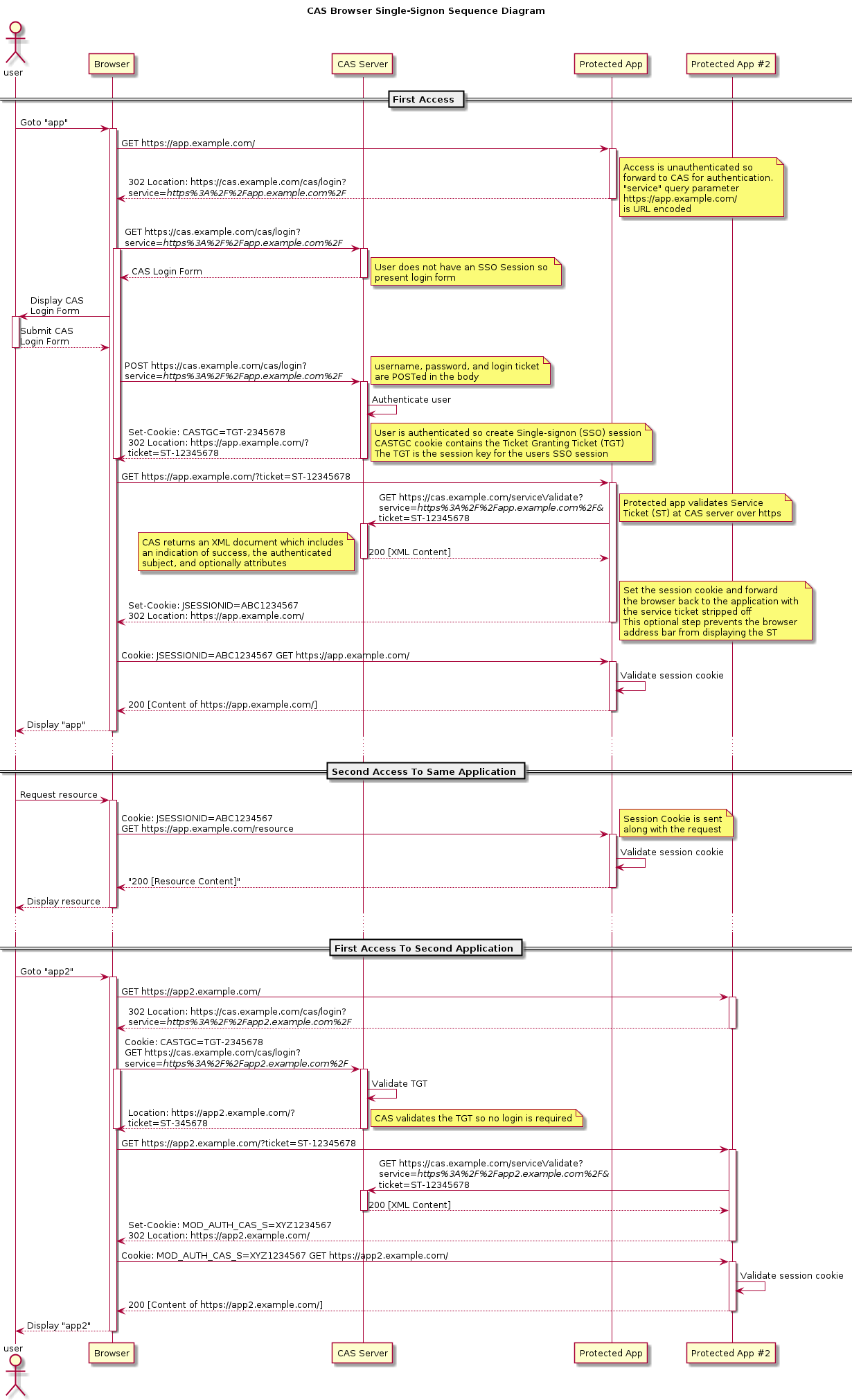
The key concepts are:

* The TGT (Ticket Granting Ticket), stored in the CASTGC cookie, represents a SSO session for a user
* The ST (Service Ticket), transmitted as a GET parameter in urls, stands for the access granted by the CAS server to the *CASified* application for a specific user.

**Versions**

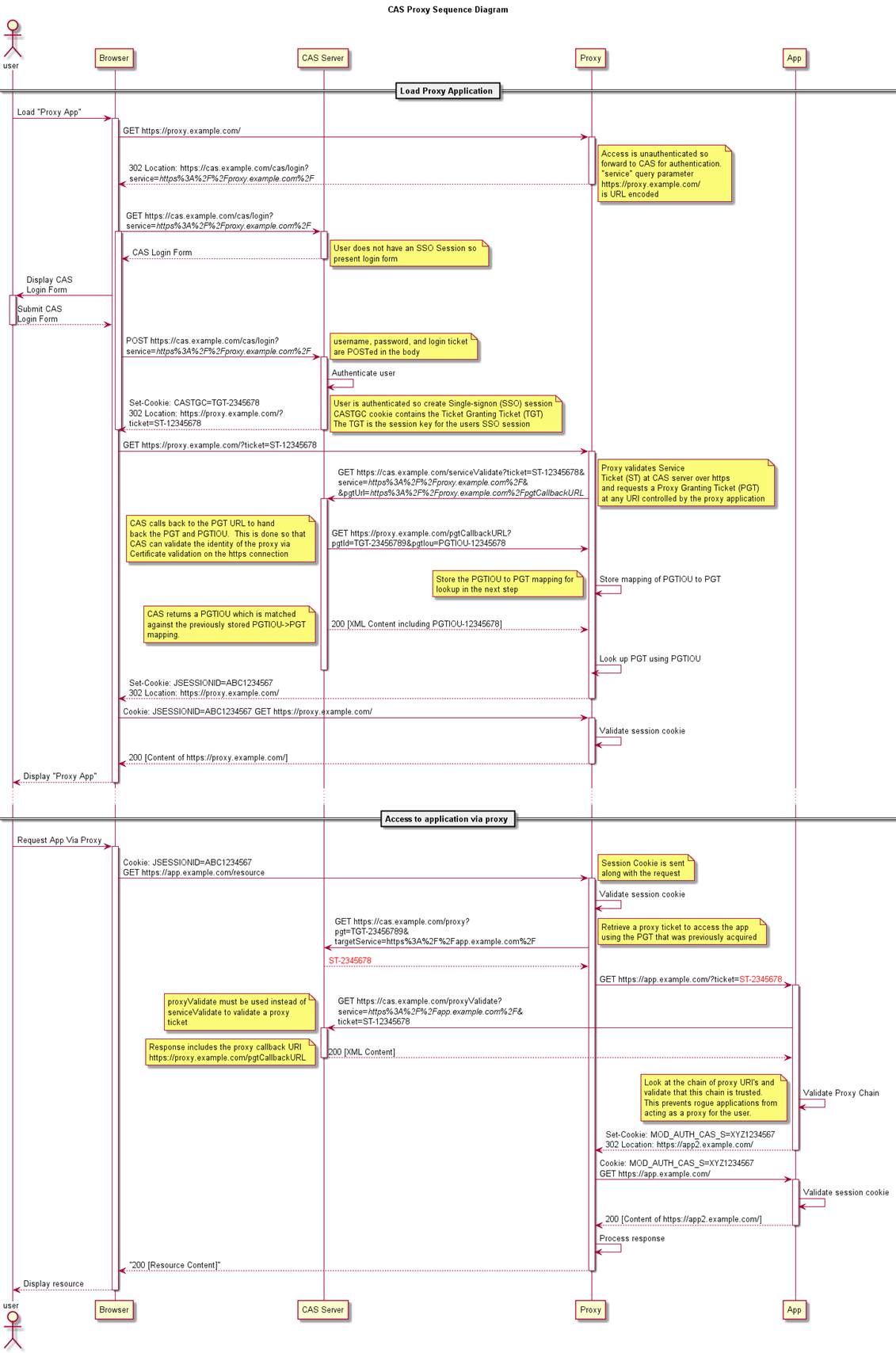
The current CAS protocol is the version 3.0. The draft version of the protocol is available as [part of the CAS codebase](https://github.com/Jasig/cas/blob/master/cas-server-protocol/3.0/cas_protocol_3_0.md), which is hereby implemented. It’s mainly a capture of the most common enhancements built on top of the CAS protocol revision 2.0. Among all features, the most noticeable update between versions 2.0 and 3.0 is the ability to return the authentication/user attributes through the new /p3/serviceValidate response (in addition to the /serviceValidate endpoint, already existing for CAS 2.0 protocol).

**Web flow diagram**



**Proxy web flow diagram**

One of the most powerful feature of the CAS protocol is the ability for a CAS service to act as a proxy for another CAS service, transmitting the user identity.



**Other protocols**

Even if the primary goal of the CAS server is to implement the CAS protocol, other protocols are also supported as extensions:

* [OpenID](http://jasig.github.io/cas/development/protocol/OpenID-Protocol.html)
* [OAuth](http://jasig.github.io/cas/development/protocol/OAuth-Protocol.html)
* [SAML](http://jasig.github.io/cas/development/protocol/SAML-Protocol.html)

**Delegated Authentication**

Using the CAS protocol, the CAS server can also be configured to [delegate the authentication](http://jasig.github.io/cas/development/integration/Delegate-Authentication.html) to another CAS server.

Source: <http://jasig.github.io/cas/development/protocol/CAS-Protocol.html>