UNICORE - authentication

Krzysztof Benedyczak
ICM, University of Warsaw

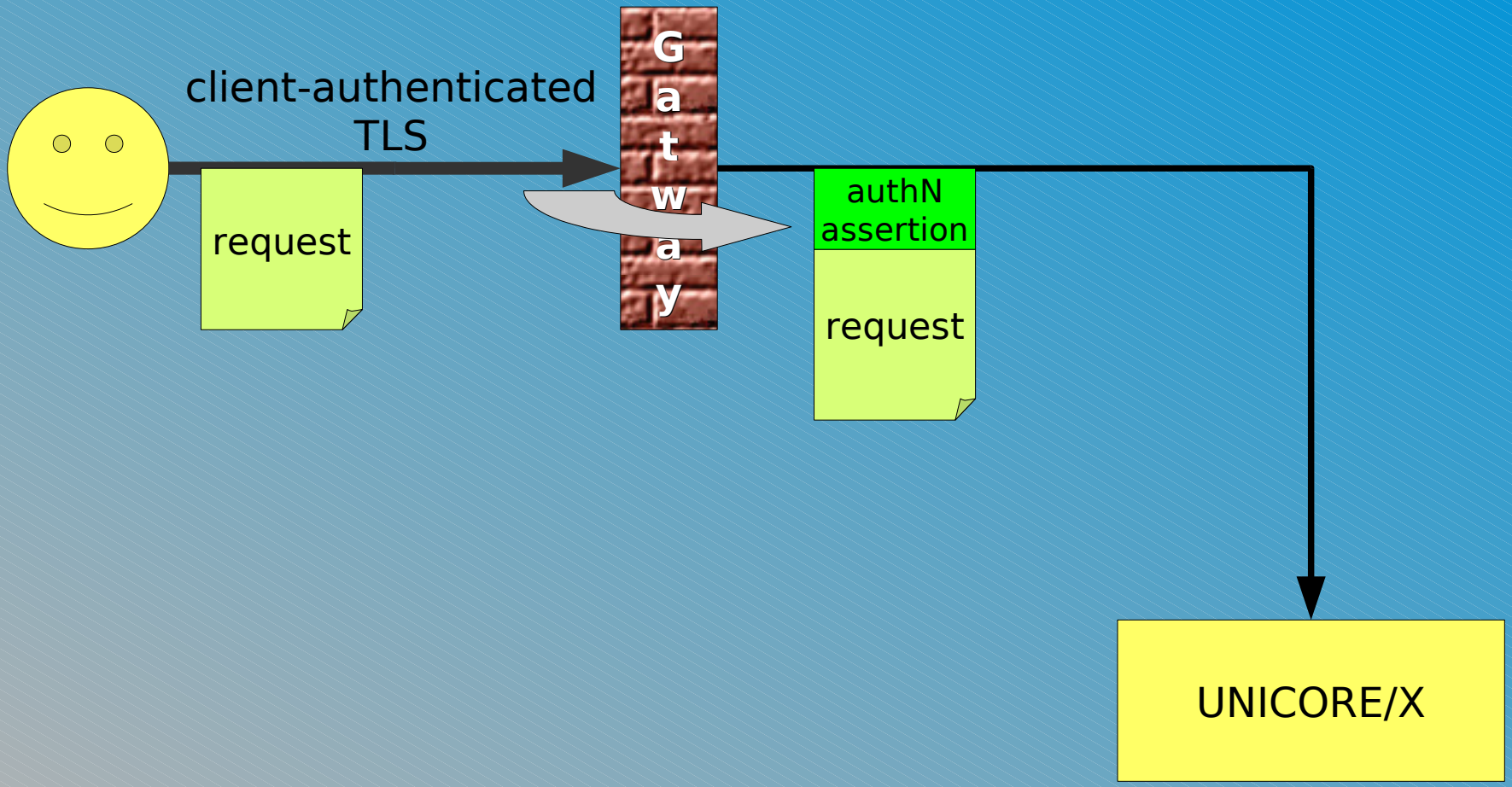
Outline

- Status of authentication in UNICORE.
- Requirements for the common authN library.

Authentication in UNICORE

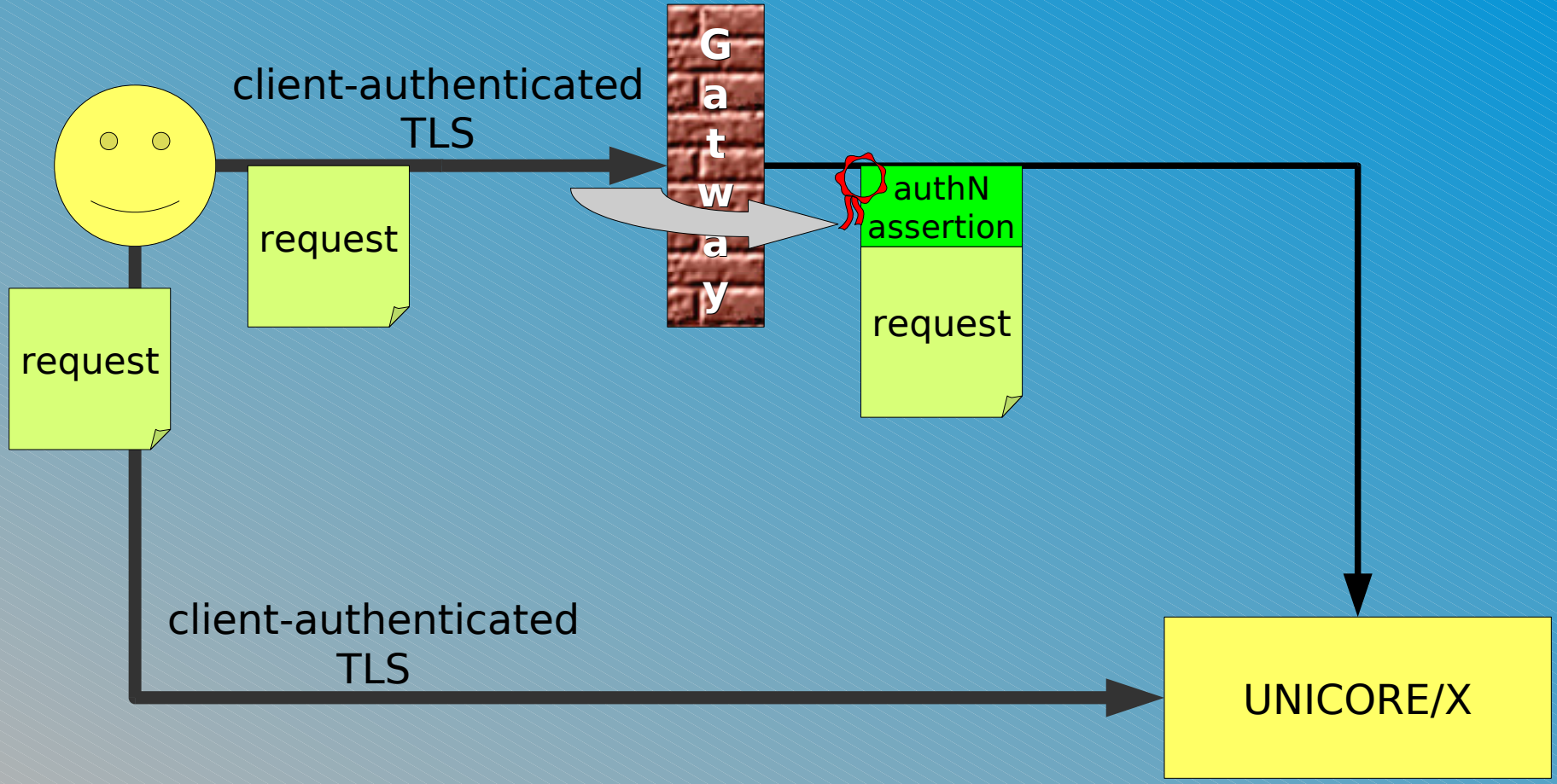
- Natively UNICORE uses the standard TLS/SSLv3 authentication with regular X.509 certificates.
- The connections are mutually authenticated.
- UNICORE does not check DNS server host name against CN in the server's certificate.
- Proxy certificates are partially supported.
  - In practice this feature is not used as Proxy certificates are unsupported in authorization stack and Proxies can not be used without UNICORE delegation assertions (ETD).

Authentication process - typical



Authentication process - direct

Authentication process - mixed



Components

- All clients (UCC, URC)
- Gateway
- Unicore/X (in direct and mixed scenario)
  - and all other servers based on wsrflite
    - wsrflite is a container used by Unicore/X providing WSRF implementation.
    - Workflow service, Service orchestrator, Registry, ...
- UVOS (not part of EMI components portfolio)
  - UVOS also uses HTTP authentication (BASIC).

Additional facts

- UVOS provides more sophisticated authentication options mostly for web browsers (SAML Authentication).
- All components (which require authentication) are written in Java.
  - Perl TSI and UDT transport libraries are out of scope.
- TLS authentication is handled by a standard Java stack with extensions (Trust Managers and Key Managers).
  - The code is in a separate library which is commonly used.

UNICORE requirements

- Java API and implementation.
- Standard TLS/SSLv3 with standard X.509 certificates supported.
- All features optional & configurable.
  - For instance host name checking can be seen as a valuable addition to UNICORE but must not be mandatory.
  - Programmatic configuration must be possible (without relaying on system-wide settings).
- HTTP auth (user&password) possible.