Single Sign-On in Service-Oriented Computing

Kurt Geihs, Robert Kalcklösch
{geihs, rkalckloesch}@ivs.tu-berlin.de

Andreas Grode
andreas.grode@dinits.de

Intelligent Networks and Distributed Systems Management
Berlin University of Technology

www.ivs.tu-berlin.de

Overview

- Motivation
- Single Sign On (SSO)
- Existing SSO Approaches
- Our SSO Prototype
- Conclusions

Motivation

- Service-oriented computing and Web services are critical ingredients for electronic commerce
- Loosely coupled services for B2B and B2C scenarios
- Open environment ➔ security
  - Effective security mechanisms
  - Lower inconvenience hurdles

Motivation (contd.)

- Actual state
  - Authentication for every service separately
  - Repeating username-password combinations
  - Password lists
- Target state
  - One authentication to the system
  - Using multiple services

Overview

- Motivation
- Single Sign On (SSO)
- Existing SSO Approaches
- Our SSO Prototype
- Conclusions

Single Sign On (SSO)

- One authentication for the access to multiple services
- Hides different account information
- SSO-System manages logins for chosen services
- Key of the kingdom
- Security of the SSO is also a major aspect

Ways of authentication

- Three major mechanisms for authentication
  - Known secret (e.g., password)
  - Material token (e.g., magnetic card)
  - Biometric attribute (e.g., fingerprint)
- No Mechanism per se better

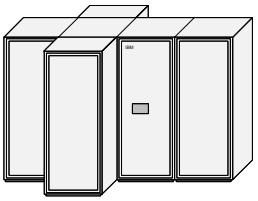
SSO Mechanisms

- Cooperative
  - Participating services know about the SSO
  - Services have to be modified
  - SSO not transparent for the services
  - Users can move directly from one service to another
- Non-Cooperative
  - Services do not know about the SSO
  - Services can be used as is
  - Not all services can be unified
  - SSO fully transparent to the services
  - Local application or server-side proxy

Overview

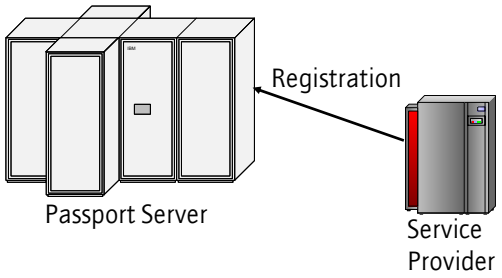
- Motivation
- Single Sign On (SSO)
- Existing SSO Approaches
- Our SSO Prototype
- Conclusions

Microsoft Passport

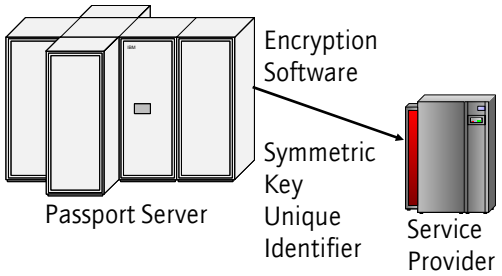


Passport Server

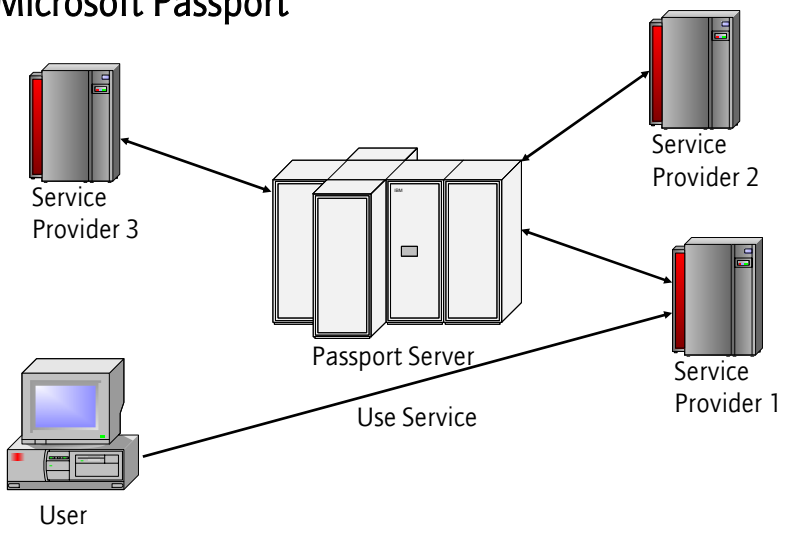
Microsoft Passport



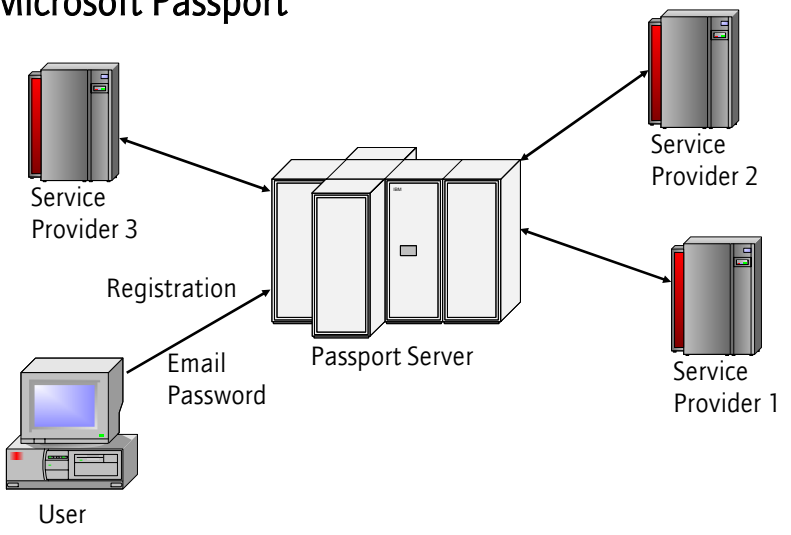
Microsoft Passport



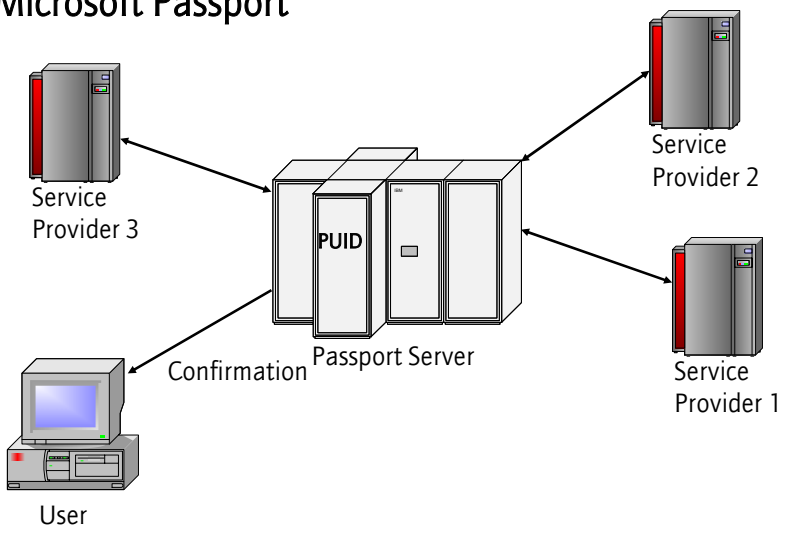
Microsoft Passport



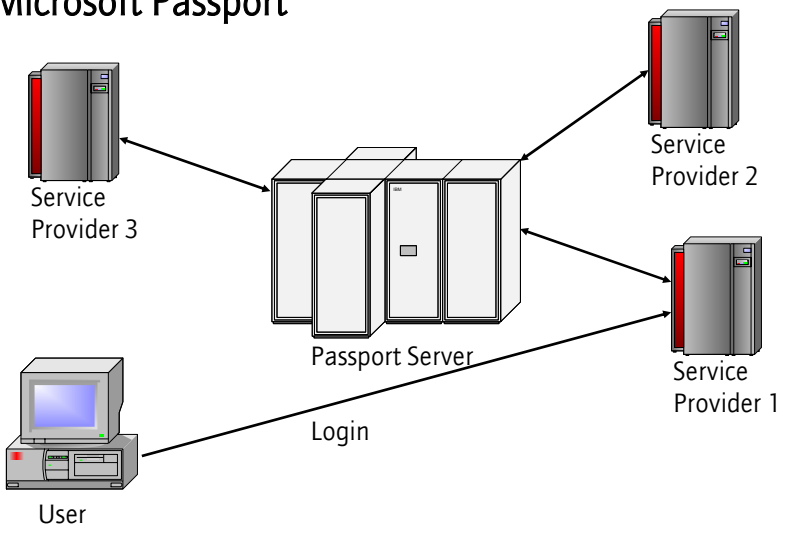
Microsoft Passport



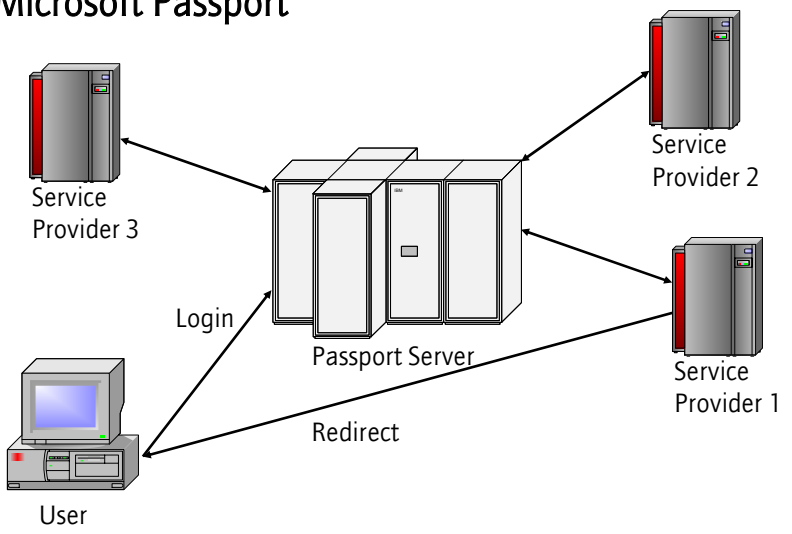
Microsoft Passport



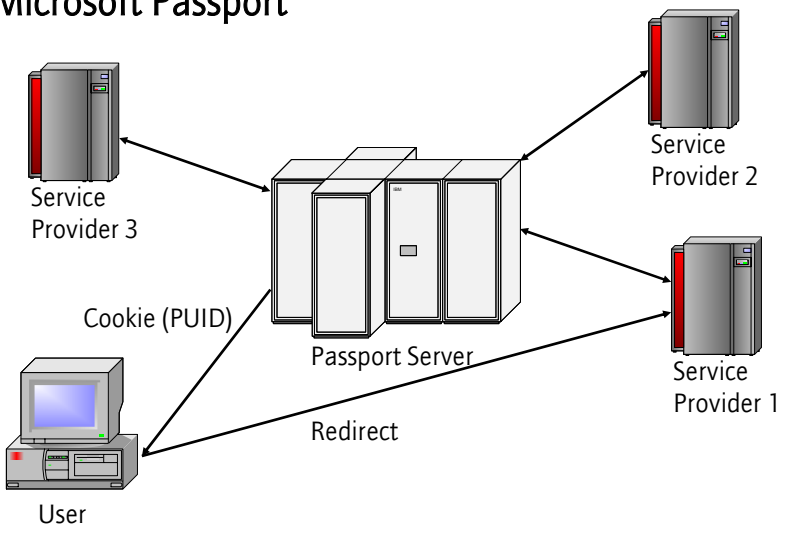
Microsoft Passport



Microsoft Passport



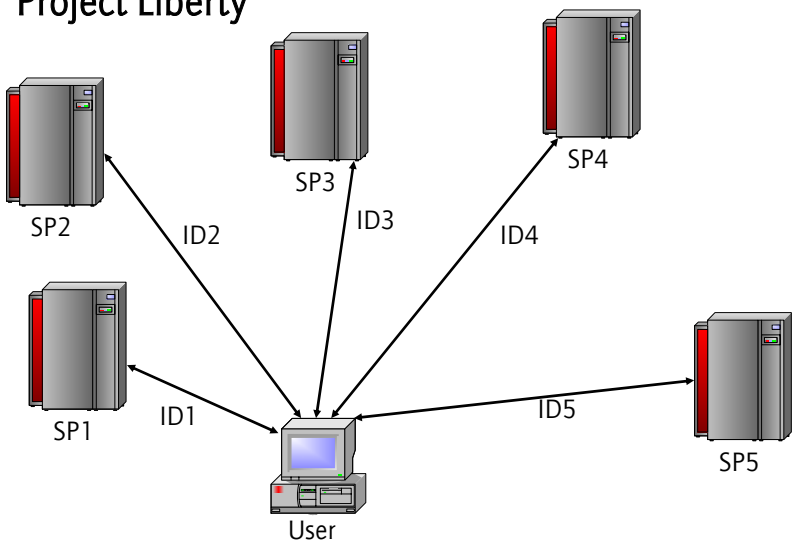
Microsoft Passport



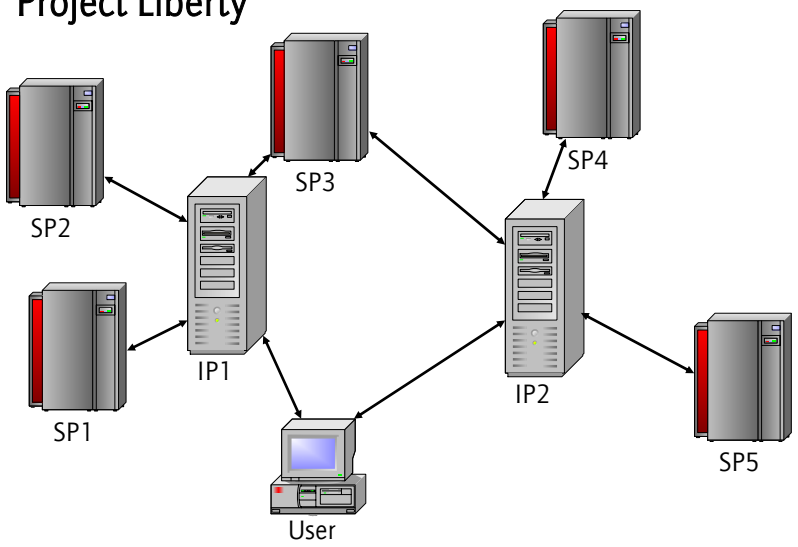
Microsoft Passport

- Centralized storage of account information
- Service provider need to be registered
- Microsoft provides symmetric key used for the communication
- Encryption done by Microsoft software installed by the provider
- User registers with valid email and password
- Passport Server manages the authentication

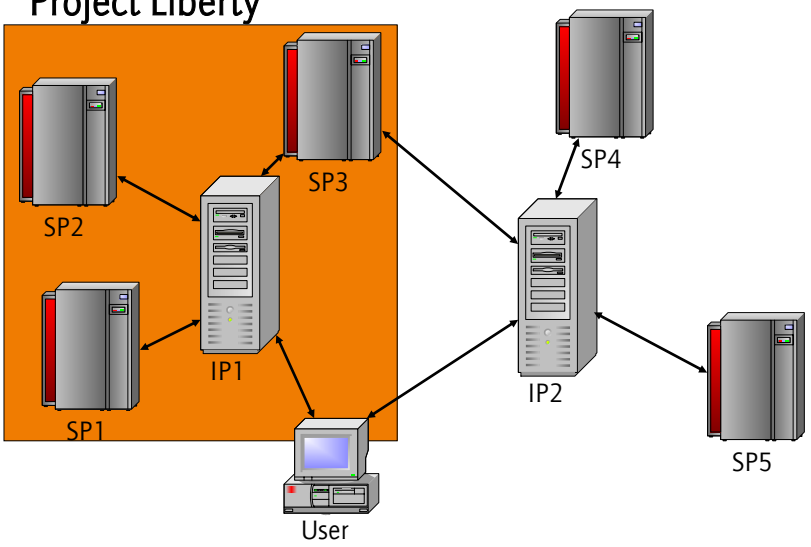
Project Liberty



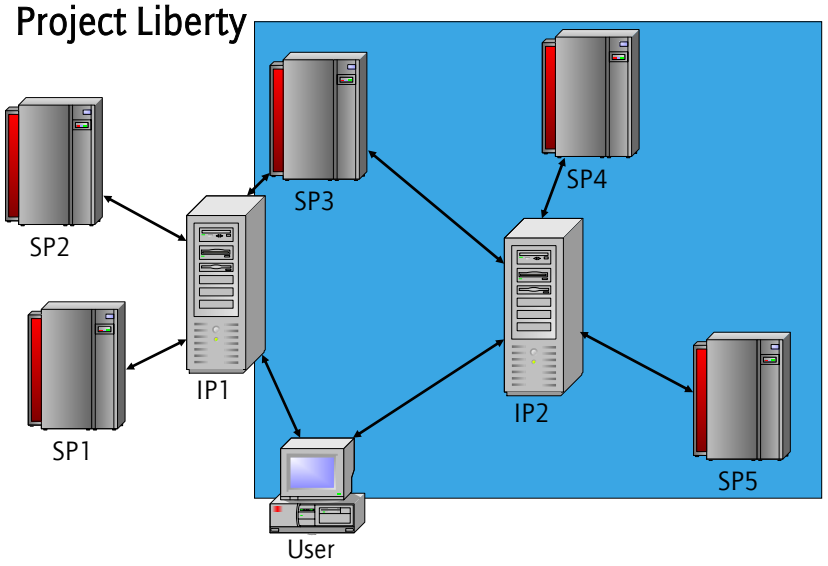
Project Liberty



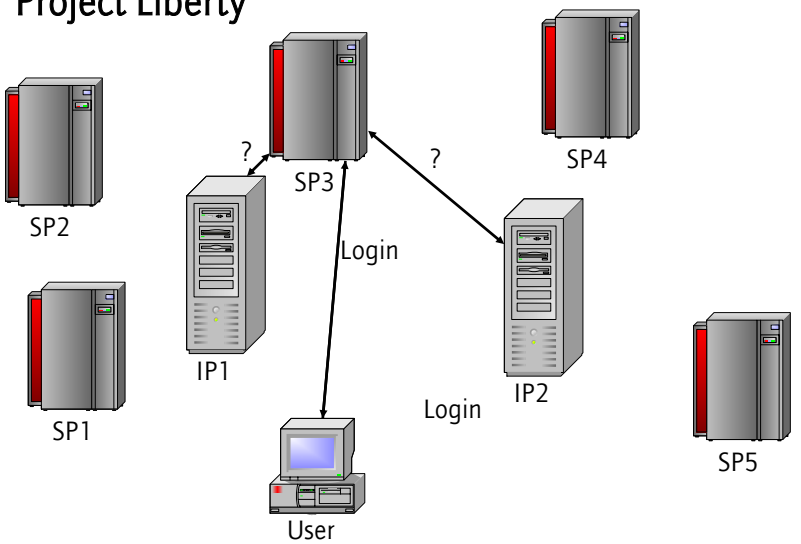
Project Liberty



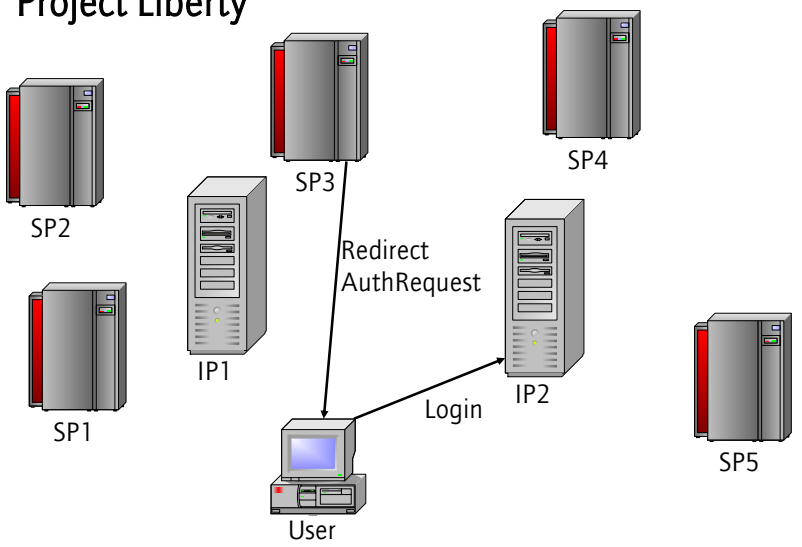
Project Liberty



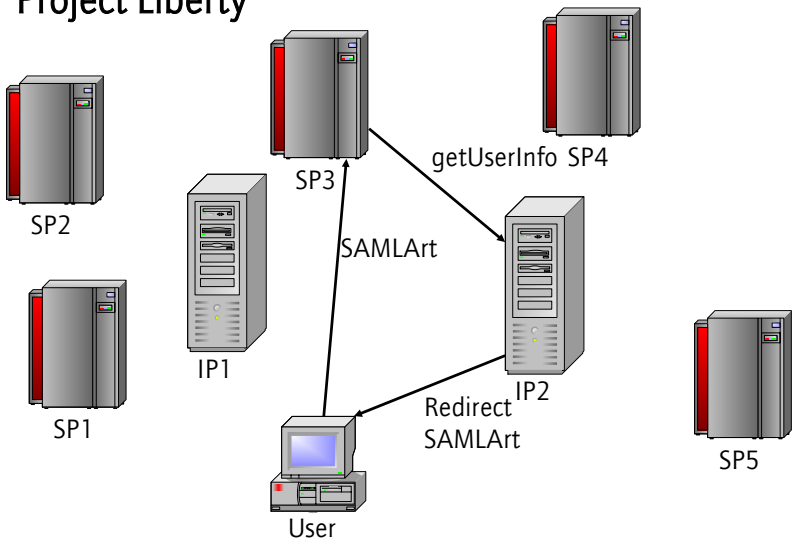
Project Liberty



Project Liberty



Project Liberty



Project Liberty

- SSO standard, not an implementation
- Coupling of distributed user identities
- Ensure compatibility and security between Liberty-aware applications
- User authentication is done by Identity Providers
- Authentication mechanism chosen by the service providers

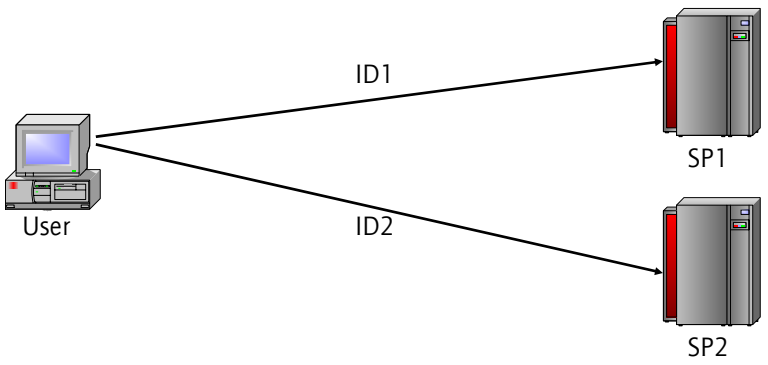
Overview

- Motivation
- Single Sign On (SSO)
- Existing SSO Approaches
- Our SSO Prototype
- Conclusions

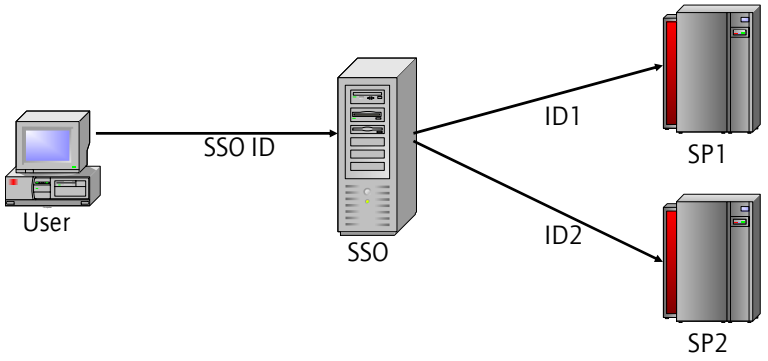
Our SSO Prototype

- WWW environment
- Non-cooperative ➔ no modification on target services
- Encapsulate remote, protected web-applications
- Service authentication mechanisms functions as before ➔ still usable without SSO
- Stores information about users and credentials for all applications

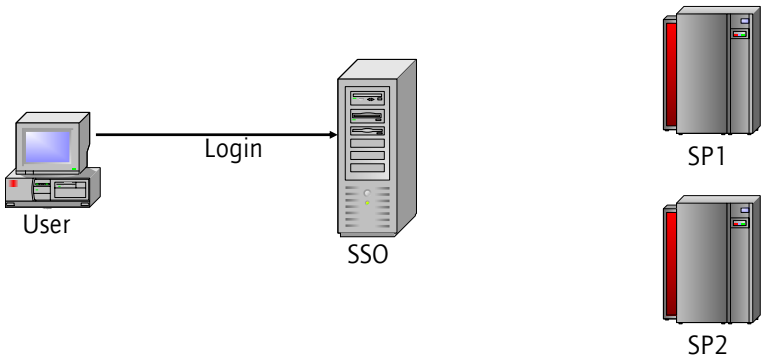
Our SSO Prototype (contd.)



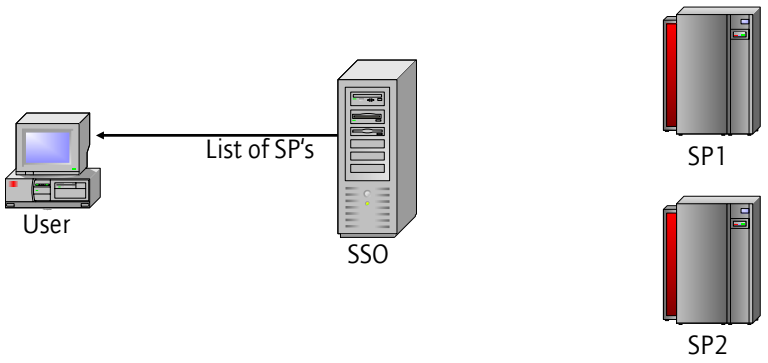
Our SSO Prototype (contd.)



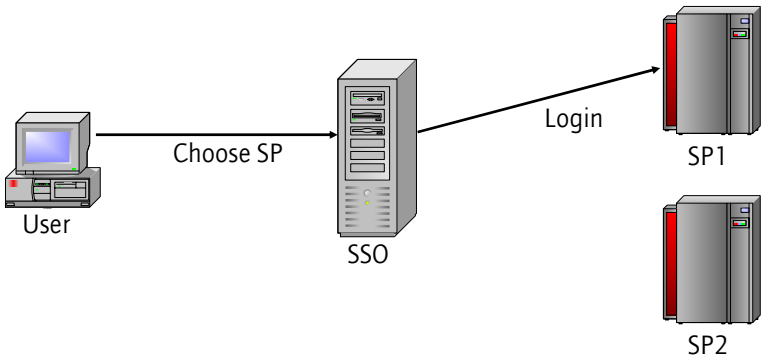
Our SSO Prototype (contd.)



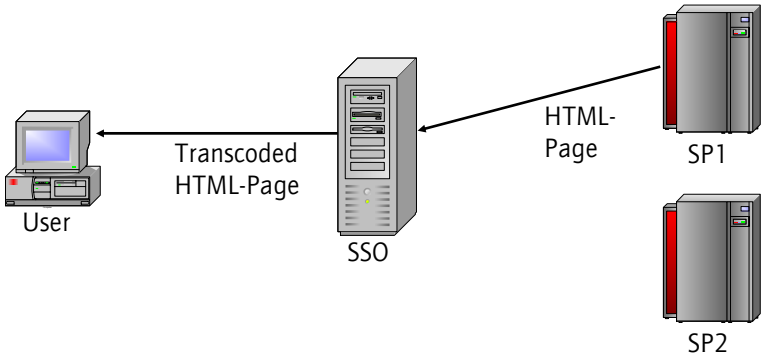
Our SSO Prototype (contd.)



Our SSO Prototype (contd.)



Our SSO Prototype (contd.)



Our SSO Prototype (contd.)

- Proxy server between user's browser and the service
- All links are transcoded
- Transcoding reverse proxy
- User authenticates only on the SSO server
- Easy to improve

Our SSO Prototype - Experiences

- Lightweight and Extendable SSO
- Service transparent SSO possible
- Currently only plain HTML
- Best used in an intranet
- Performance has to be improved
- Reliability and scalability are open issues
- Authentication mechanism exchangeable
  ➔ making applications Liberty-aware

Conclusions

- Non-Cooperative SSO possible (in WWW)
- No modification on services needed
- Possible to make Non-Liberty-aware applications Liberty-aware

Thank you.

Question and comments are welcome.

Robert Kalcklösch
rkalckloesch@ivs.tu-berlin.de
Intelligent Networks and Distributed Systems Management
TU Berlin
www.ivs.tu-berlin.de

Telecommunications Institute
Faculty IV – Electrical Engineering & Computer Science
TU Berlin

phone: +49 30 314-79834
fax: +49 30 314-24573

office@ivs.tu-berlin.de

Secretary EN 6
Einsteinufer 17
EN-Gebäude

D-10587 Berlin
Germany