

28 February, 2007



# A Lightweight Service Grid based on Web Services and Peer-to-Peer

Markus Hillenbrand, Joachim Götze, Ge Zhang, Paul Müller

Markus Hillenbrand  
University of Kaiserslautern, Germany  
Integrated Communication Systems Lab  
email: [hillenbr@informatik.uni-kl.de](mailto:hillenbr@informatik.uni-kl.de)

KiVS 2007  
Bern, Switzerland



# Outline

- Background
- Venice Architecture
  - Security Domains
  - Management Services
  - Information Services
  - Application Services
- Performance Evaluation
- Conclusion and Future Work



## What is a Grid?

- Len Kleinrock, 1969:

“... the spread of computer utilities, which, like present electric and telephone utilities, will service individual homes and offices across the country“

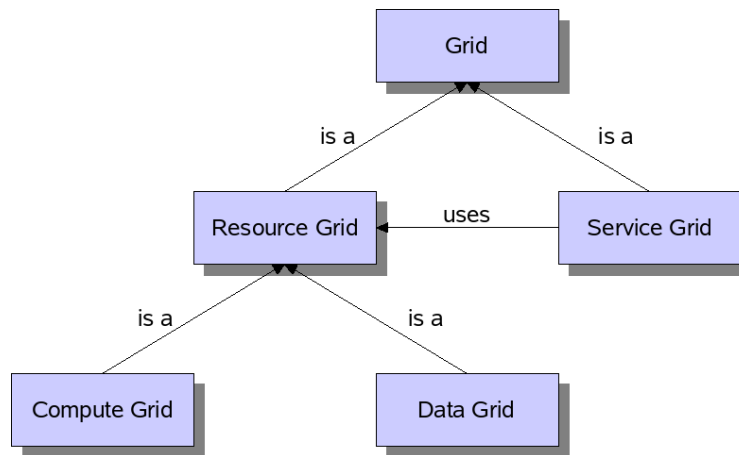
- Ian Foster, 2002

“... coordinates resources that are not subject to centralized control using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service”

<http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf>



# A Grid Classification



- **Compute Grids**  
provide access to CPU cycles  
→ e.g. clusters
- **Data Grids**  
provide access to large data storage → e.g. data bases
- **Resource Grids**  
provide access to resources  
→ e.g. special hardware
- **Service Grids**  
provide functionality by combining resources with algorithms



# What is a Lightweight Service Grid?

- A Service Grid ...
  - Abstracts from underlying hard- and software
  - Hides the complexity of its resources
  - Focuses on the end-user
  - Provides intuitive means for access
- A Lightweight Service Grid ...
  - Is easy to deploy  
(i.e. quick service creation and deployment)
  - Is easy to maintain  
(i.e. tool support for management, monitoring, and configuration at runtime)
  - Is easy to use  
(i.e. no installation for end-users, GUI for access)

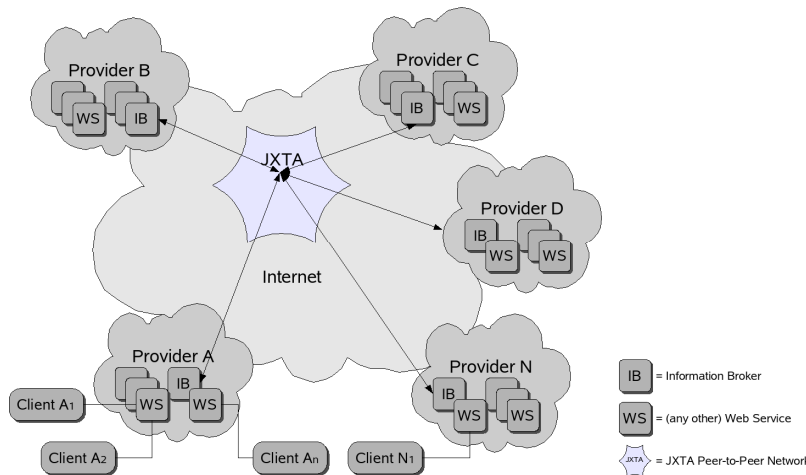


# What is the Venice Service Grid?

- A software infrastructure:
  - Based on a service-oriented architecture (SOA)
  - With focus on openness, dependability and security
  - Service deployment on the Internet
- A set of services:
  - Service management at runtime
  - Service information and access
  - Service collaboration and communication
  - Services for building applications upon
- A runtime environment
  - For service development and deployment
  - For client development and service access



# Top Level View

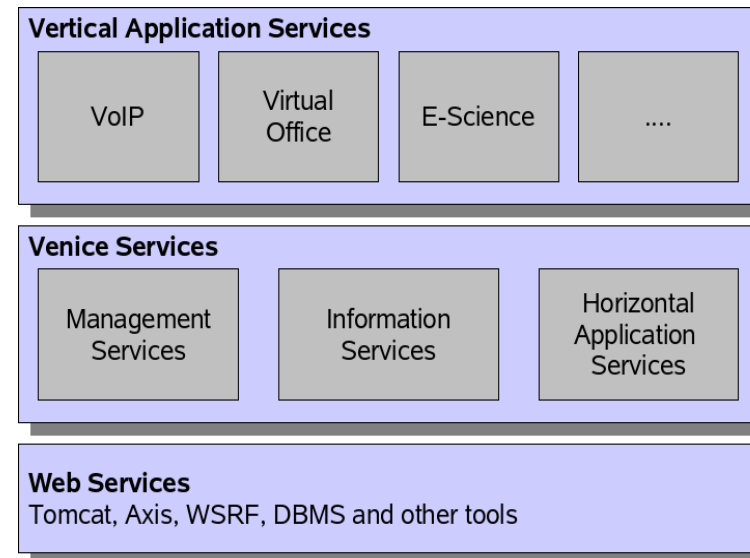


- Service providers are
  - Independent
  - Autonomous
- Service requestors
  - Have a home domain
  - Can use services of other domains
- Venice is responsible for
  - Mutual authentication
  - Distributed authorization
  - Service brokering
  - Service access



# A Venice Service Provider

- Management Services
  - User Management
  - Service Management
- Information Services
  - User data
  - Service data
  - Cross-domain communication
- Application Services
  - Horizontal:  
for all application domains
  - Vertical:  
for a special application domain





# Management Services

- Single Sign-on (SSO)
  - Authentication
  - Authorization
  - Domain handling
- Accounting (MAB)
  - Metering
  - Accounting
  - Billing
- Preferences
  - Unified interface
  - Runtime configuration
- Software (SDS)
  - Deploy software
  - Versioning
  - For Clients and Providers



# Information Services

<http://www.icsy.de>

- Domain
  - Meta data
  - Contact
  - Bootstrap services
- Services / Resources
  - Service broker
  - Meta data
  - Mutual Authentication
  - Client software
- Users
  - Contact means
  - Online status
  - Arbitrary properties
- Events
  - Notification framework
  - Source, Sink, Proxy, Filter
  - Publish/Subscribe
- Federation
  - Service brokering via P2P
  - Role mapping



# Application Services

- Data storage
  - Property storage
  - File system
  - Time keeper
- Communication
  - Email
  - SMS
  - Voice over IP
  - Presence
- Miscellaneous
  - Location Service
- Application Scenarios
  - Voice over IP
    - Basic VoIP functionality
    - H.323
    - SIP
    - Supplementary Services
  - Virtual Office
    - Print
    - Scan
    - Store
  - E-Science
    - Call for Papers

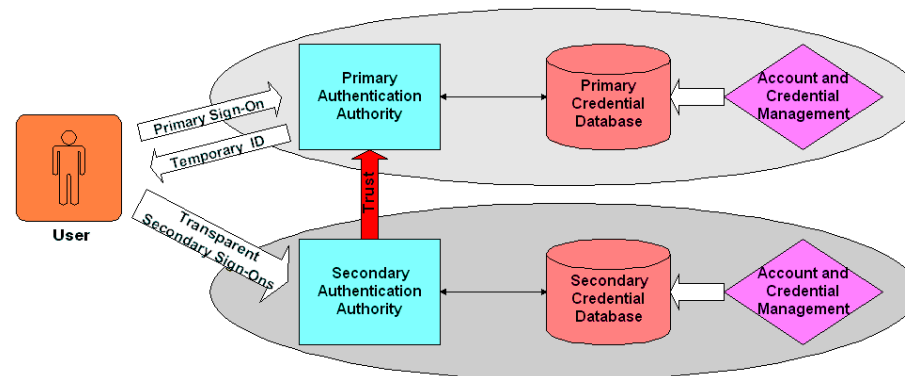


# Single Sign-on Service (SSO)

<http://www.icsy.de>

“Single sign-on is the ability for a user to authenticate once to a single authentication authority and then access other protected resources without re-authentication.”

- Authentication & authorization
  - Based on tokens
  - Role based access control
  - Enables federations of trust
  - Role mapping across domains

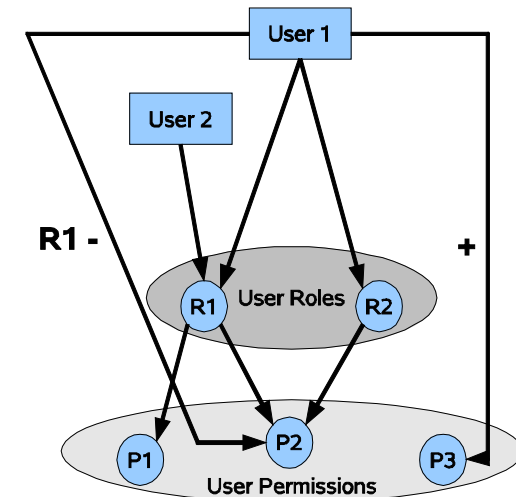


By Jan de Clercq in “Single Sign-On Architectures”, 2002



# Single Sign-on Service (SSO)

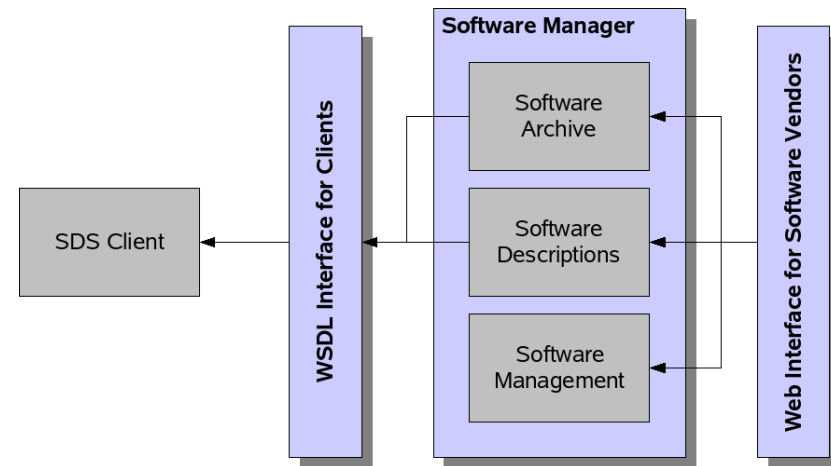
- Authentication
  - Kerberos-like (mutual) authentication
  - Password never on the network
  - Resilient against replay attacks
- Authorization
  - Fine grained access control
  - No central authorization policy necessary
  - Delegation mechanism
  - Direct and indirect trust





# ▶▶▶ Software Deployment Service (SDS)

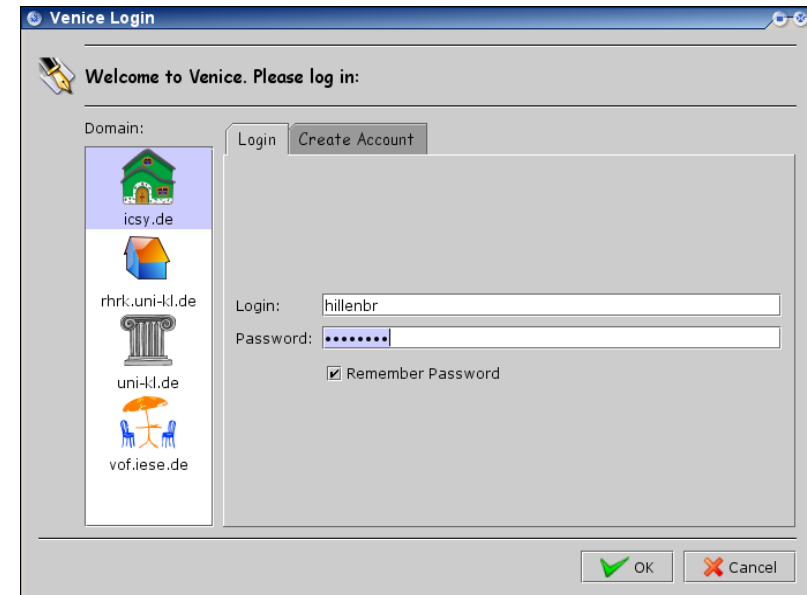
- SDS allows for:
  - different computer architectures
  - different operating systems
  - recursive dependencies, i.e.  $A \Rightarrow B$  and  $B \Rightarrow C$  then  $A \Rightarrow C$
  - versioning/updating ( $A \Rightarrow C_1$  and  $B \Rightarrow C_2$ )
  - little (autonomous: no) user interaction





# Domain Information Service (DIS)

- General starting point
  - SSO Service
  - Information Broker
  - Notification Service
  - WHOIS Service
  - Resources
  - Services
- Provides domain info:
  - Name
  - Description
  - Administrative Help





# Information Broker (IB)

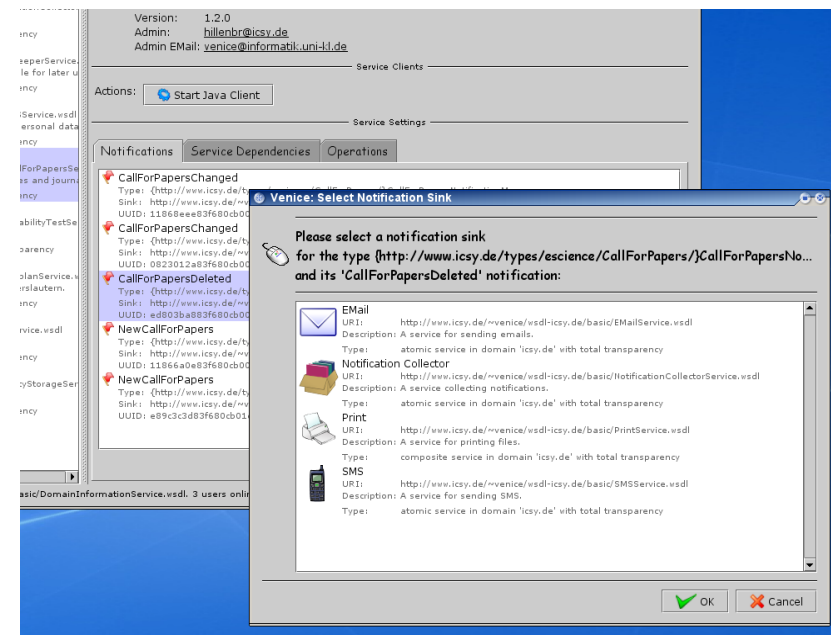
<http://www.icsy.de>

- Finds in other domains
  - DIS
  - Resources
  - Services
  - User data
- Communicates with other IBs via JXTA
  - Always up-to-date
  - No centralized registry
  - P2P outside the domain
  - Web service in domain



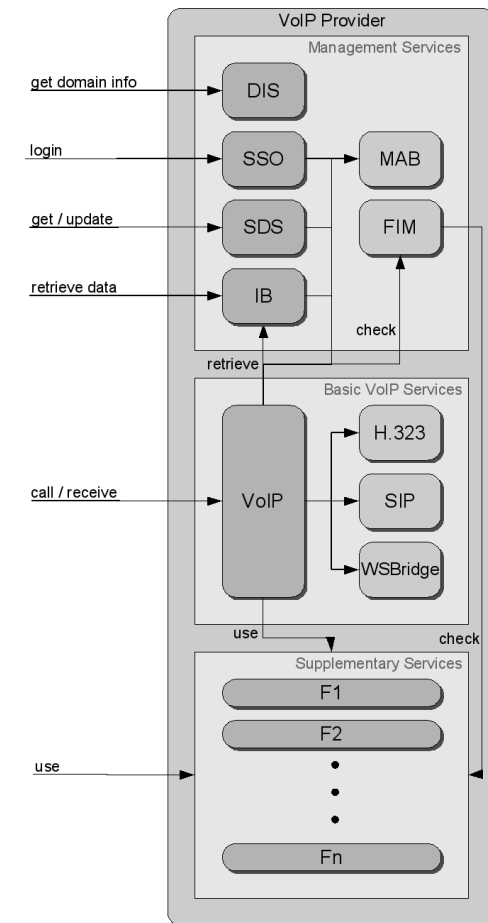
# Notification Services

- Notification Source
  - Issues messages
  - Supports message types
- Notification Sink
  - Receives messages
- Notification Proxy
  - Forwards messages
  - Filters messages
- Notification Service
  - Publish/subscribe mechanism
  - Distributes messages



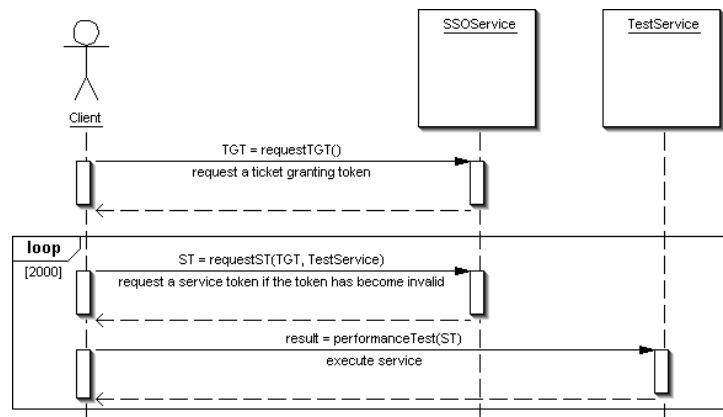
# Voice over IP Services (VoIP)

- Basic VoIP functionality
  - Initiate and control RTP transmission
  - Gateway services for H.323 and SIP
  - ISDN via SIP and Asterisk
- Supplementary services
  - Selectable by user and administrator
  - CFU, CFBY, CFNR, CLIP, CLIR, COLP, COLR, CCNR, CCBY, TCS, Voice Mail, Call Center, Address Book
  - Feature Interaction Manager (FIM)
- Simple (= thin) software client
  - Adaptive GUI according to supplementary service selection
  - Based on Java and JMF



# Performance Evaluation Setup

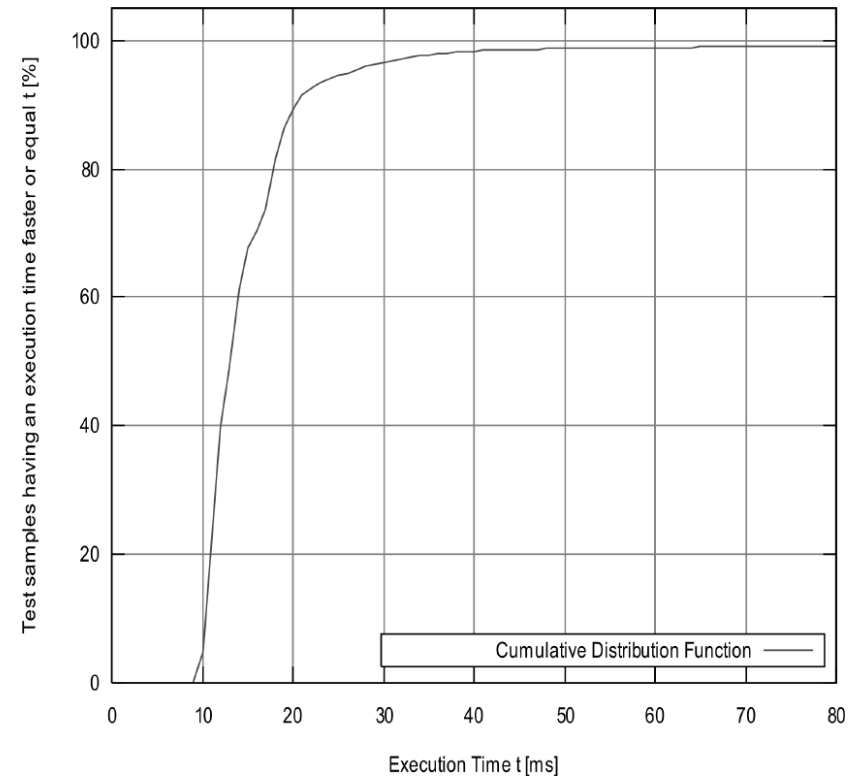
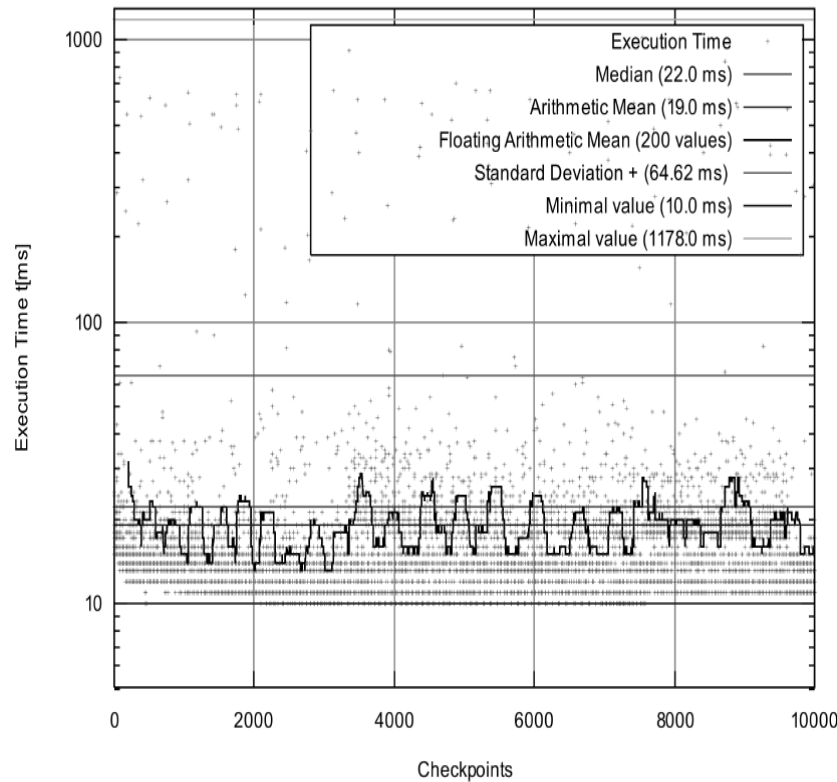
- Hardware
  - 1 x Server: 2xP4, 3GHz
  - 5 x Client: 1xP4, 2.4-3 GHz
  - Network: 100 Mbit/s LAN
- Software
  - OS: Linux (SuSE 10.1)
  - Java: JDK 1.5 from Sun
  - Tomcat 5.5 with Axis 1.4
- Time Keeper Service
  - Purpose
    - Measure
    - Store
    - Evaluate
  - Useful for
    - Development
    - Runtime
    - Scalability considerations
  - Evaluates
    - Floating / arithmetic mean
    - Standard deviation
    - Cumulative distribution



# Performance Evaluation Results

<http://www.icsy.de>

Average Execution Time: 19.0 ms (+/- 45.62 ms)





# Conclusion and Future Work

- Conclusion
  - Venice is a lightweight Service Grid
  - Venice provides
    - management,
    - information, and
    - application services
  - Venice defines
    - a software infrastructure
    - a runtime environment
- Future Work
  - Extensive dependability support
  - WSRF compatibility (e.g. Globus Toolkit 4)
  - Additional services

28 February, 2007



Thank you for your attention.

Questions?