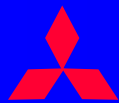


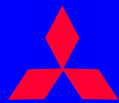
Security and Reliability in Concordia™

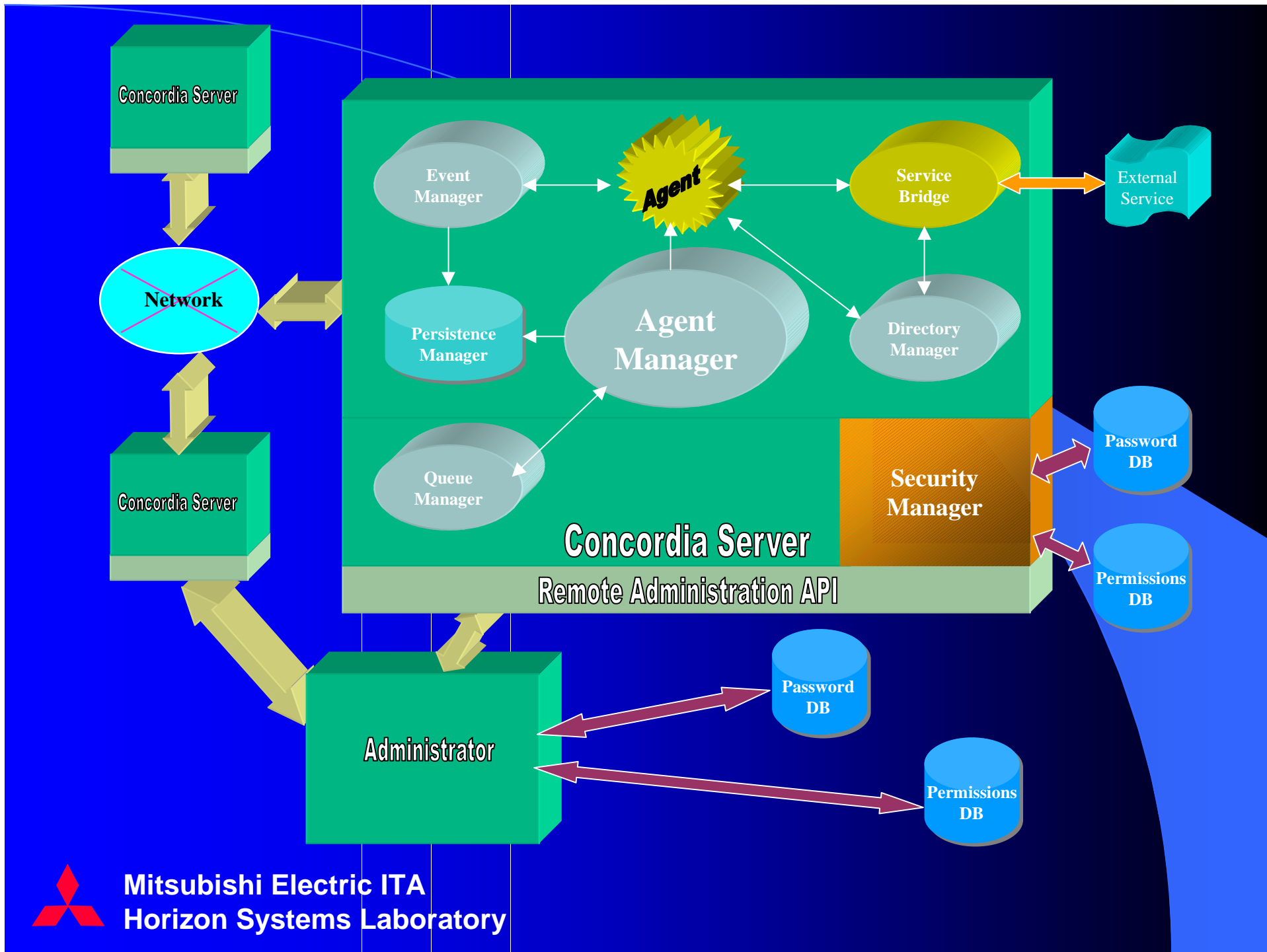


Mitsubishi Electric ITA
Horizon Systems Laboratory

Concordia System Architecture

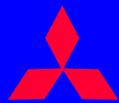
- Concordia system provides support for flexible agent mobility, agent collaboration, persistence of agent state, reliable agent transmission, and security.
- Consists of a set of Java class libraries for server execution, agent application development, and agent activation.
- Consists of a number of interacting *managers* and objects executing in a common Java VM realized as the *Concordia Server*.





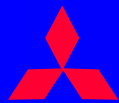
Concordia Security

- Agent Protection
 - Protects agents and the information they carry from illegal access or tampering.
 - Utilizes encryption and signing technology to guarantee agent safety.
- Resource Protection
 - Protects server resources from attack by agents.
 - Utilizes a user identity paired with resource access control lists.



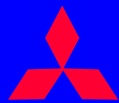
Agent Protection

- Transmission Protection
 - Protects an agent while in transit
- Storage Protection
 - Protects an agent while stored in an on-disk representation such as the Concordia persistent store
- Both use a combination of symmetric and public-private key encryption to provide safety and good performance
- Available in next Concordia release scheduled for 2nd half of 1998



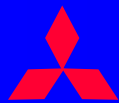
Resource Protection

- Agent's access to server resources based on the *identity* of the user who launched the agent
- Different than the traditional *sandbox* or *code signing* models



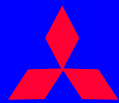
User Identities

- Every agent has an identity
- Identifies user who launched the agent
- Composed of a username, a group name and a (*hashed*) password (*john@accounting*)
- Identity is validated at each stop in an agent's travel against a server-specific *password file*
- Special Identities
 - **untrusted** - for unverifiable or non-secure agents
 - **default** - defines default security policy



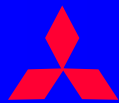
Resource Permissions

- Control access to system resources
 - Like ACL's
- Exist for standard Java resources
 - Like filesystem or network access (*File.read*)
- Exist for agent-specific *resources*
 - Such as the ability to arrive at a server, launch other agents, or receive Concordia events (*Agent.arrive*)
- Can be assigned or denied to users or to groups of users via *permissions file*



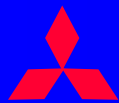
Queue Manager

- Provides for reliable transmission of agents in unreliable networks via 2PC handshaking.
- Provides for “store and forward” operation.
- Design goals centered on achieving optimal disk space utilization and fast write operations.
- Utilizes combined on-disk log-based file structure for message data and log records.



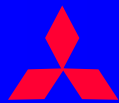
Persistence

- Concordia Servers may be configured to save their internal state to persistent storage
- The Agent Manager may write agents' states to persistent storage
 - Makes it possible to restart an agent after a Concordia Server fails and restarts
 - Idempotency is required unless agents checkpoint their internal state to persistent storage
- System administrators may enable persistence for the Concordia Server on a per-component basis



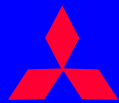
Proxies

- Concordia's proxies shield agents from the effects of server and system failures
- Proxies transparently re-establish long-lived connections after failures:
 - Event Manager Proxy
 - Proxies for distributed objects which may be remote



Conclusion

- Concordia's security model provides
 - Protection of server resources from attack by agents
 - Protection of agents from tampering during transmission or when stored on disk
- Concordia reliability features:
 - Enable reliable transmission via 2PC handshaking
 - Re-establish network connections after failures
 - Provide more fault tolerant communication and operations
 - Restore agent and server state after failure or shutdown



Wrap Up

- Concordia V1.0 available now
- Next Concordia release scheduled for 2nd half of 1998
- More info
<http://www.meitca.com/HSL/Projects/Concordia/>
- Questions?

