Secure Virtual Enclaves

February 4, 2000

Deborah Shands, Richard Yee

Jay Jacobs, E. John Sebes

Outline

- **Project Overview**
- SVE Architecture
- Observations
- Results/Conclusions

Coalition Examples

- *Commercial*: outsourcing, contractors, or customers needing limited access to corporate data
- *Civilian*: disaster/incident response teams and crisis management
- *Military*: joint task forces engaged in distributed collaborative planning

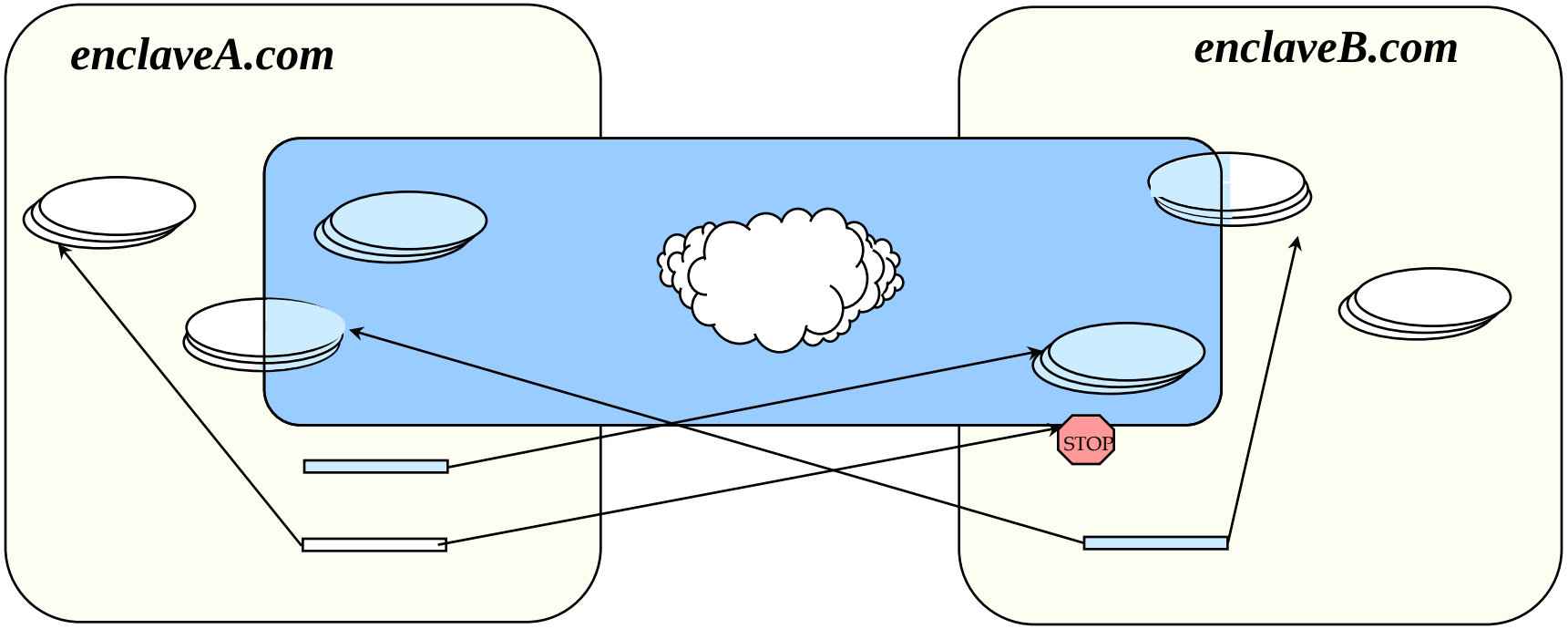
SVE Project Goals

- Support collaborative computing
- Provide mechanisms to control sharing
- Enable unified approach to multiple distributed application technologies (e.g., Java, DCOM, web apps.)
- Support dynamic access policies, allowing changes to: SVE membership, resources to be shared, and access types permitted

SVE Project Constraints

- Ensure application transparency
- Retain organizational autonomy over local resources
- Use only standard network protocols
- Use only commercially available operating systems

Concept of Operation



Legend:

Services in SVE
Services partly in SVE
Services not in SVE
Principals in SVE
Principals not in SVE

NAI LABS
The Security Research Division of Network Associates, Inc.

SVE Concept of Operation

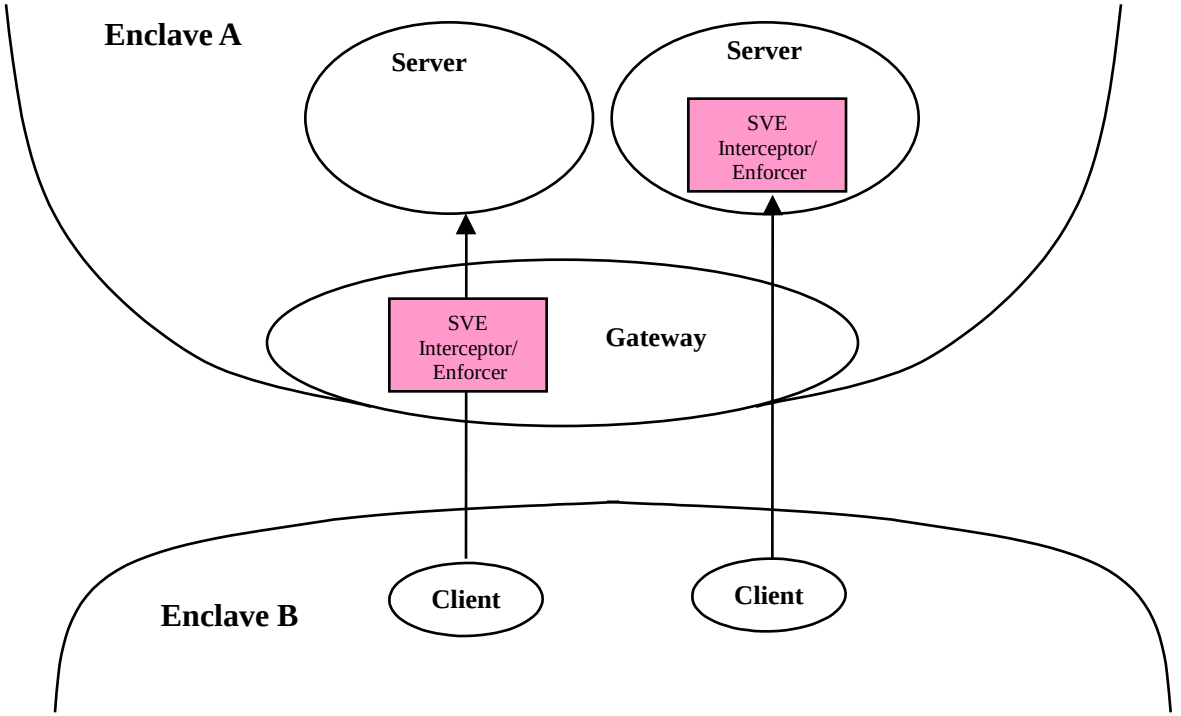
- Virtual enclave: formed by collaborators sharing resources and services
  - Enclaves define limited trust relationships with one another
  - Each enclave specifies internal resources accessible to partners
- Secure virtual enclave: each enclave's exports are
  - Protected from access by non-SVE members
  - Available to SVE members as specified by access policy
- Dynamic modification: automatic reconfiguration due to changes in SVE membership, resources, access policy

NAI LABS
The Security Research Division of Network Associates, Inc.

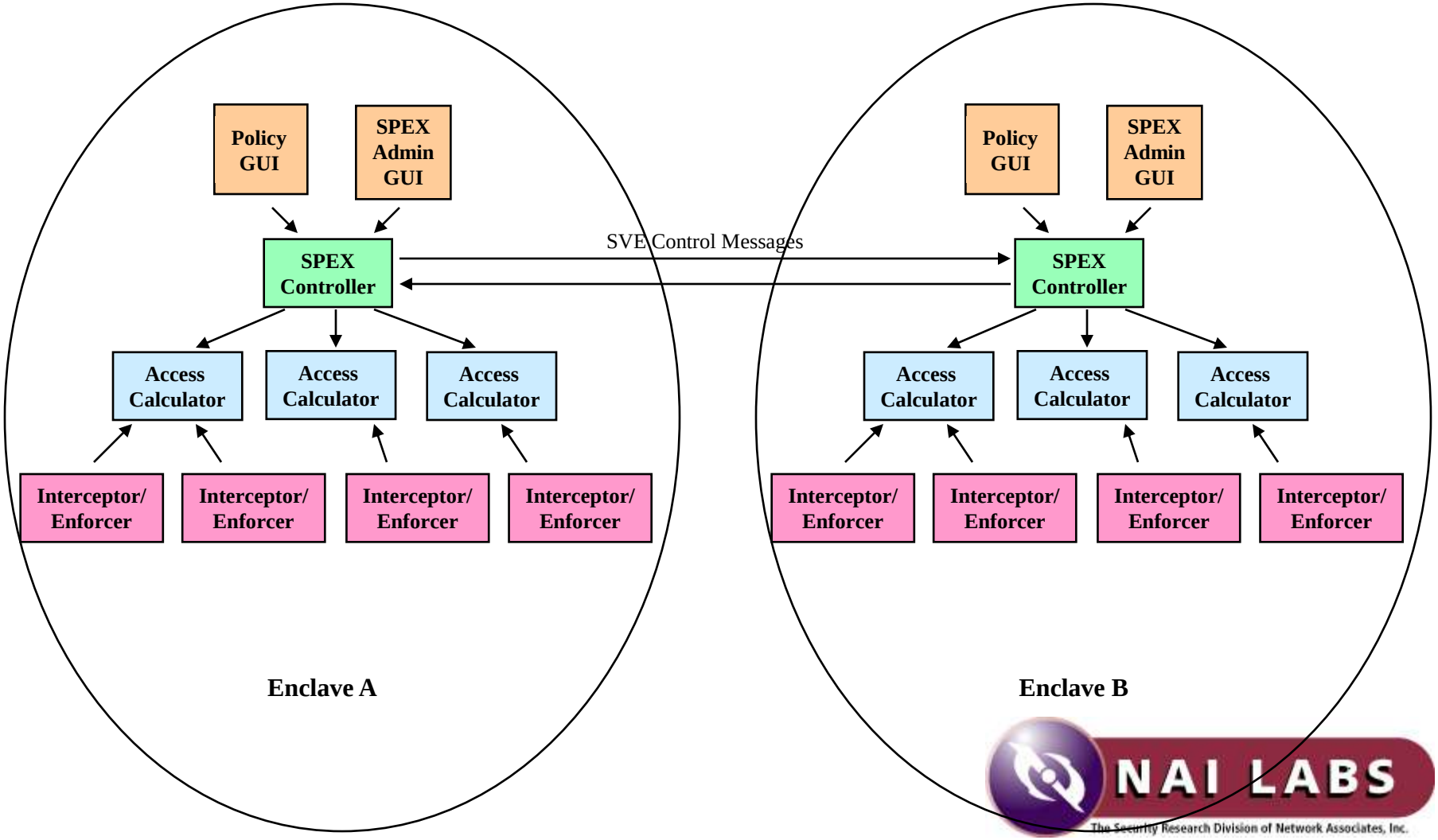
Outline

- Project Overview
- **SVE Architecture**
- Observations
- Results/Conclusions

Client-Server Architecture



SVE Infrastructure Architecture



NAI LABS
The Security Research Division of Network Associates, Inc.

SVE Policy Semantics

- Current SVE policy semantics are very similar to Object-Oriented Domain and Type Enforcement (OODTE)
- Principals are mapped to a **domain** equivalence class using a set of domain derivation rules
- Resources are mapped to a **type** equivalence class
- **Access matrix** is formed by associating a set of types with a given domain
- **Principal recognition rules** are domain derivation rules that are published by an SVE member to allow its principals to be recognized by other SVE members

NAI LABS
The Security Research Division of Network Associates, Inc.

Outline

- Project Overview
- SVE Architecture
- **Observations**
- Results/Conclusions

Enclave Autonomy

- Organizations require a certain level of autonomy
- Autonomy is a difficult requirement for distributed security systems
- SVE system supports autonomy
  - Most components of access policy used only within the local enclave
  - An enclave may unilaterally withdraw from an SVE at any time
- Need to balance autonomy and collaboration requirements via business decisions

Ambiguous Policy Semantics

- Meaning of policy statements known only within defining enclave (e.g., “manager” role)
- How to prevent misunderstandings as coalitions are formed???
  - Establish semantics offline
  - Represent and negotiate semantics within system

Outline

- Project Overview
- SVE Architecture
- Observations
- **Results/Conclusions**

SVE Prototype Results

- Supports coalition sharing
- Supports dynamic changes to both coalition membership and resource access policies
- Supports enclave autonomy
- Provides experimental platform for studying security policies for distributed systems