A High-Confidence Broker of Security Services

Tim Sauerwein
Galois Connections Inc.
March 2002

With contributions by:

Brian Huffman
Andy Gill
Brett Letner
Laura McKinney
Peter White

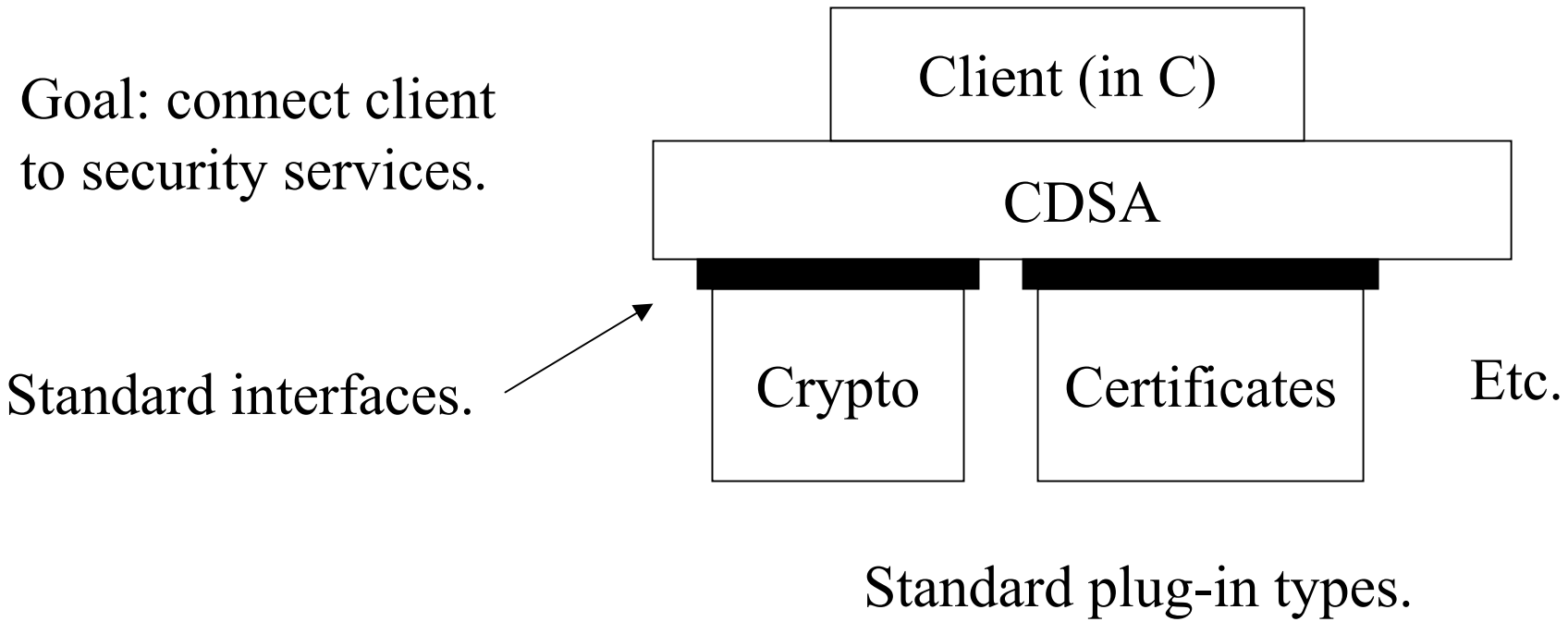
Outline

- **What is the CDSA?**
- What is the H-CDSA?
- Selected technical highlights:
  - Secure IPC on Linux
  - Remote Procedure Call
  - Trust Policies

The CDSA

"Common Data Security Architecture"

Goal: connect client to security services.



Some CDSA Features

- CDSA and plug-ins loaded into the client's address space.
- CDSA tries to isolate client from plug-ins.
- Client, CDSA, and plug-ins can check authenticity of themselves and of each other ("bilateral authentication").

CDSA Standard Plug-in Types

- Cryptographic Service Provider
- Certificate Library
- Trust Policy Module (mostly certificate-related operations at a higher level)
- Data Storage Library
- Authorization Computation Module (checking permissions based on access control lists and certificate chains).
- Elective Module (add your own module type)

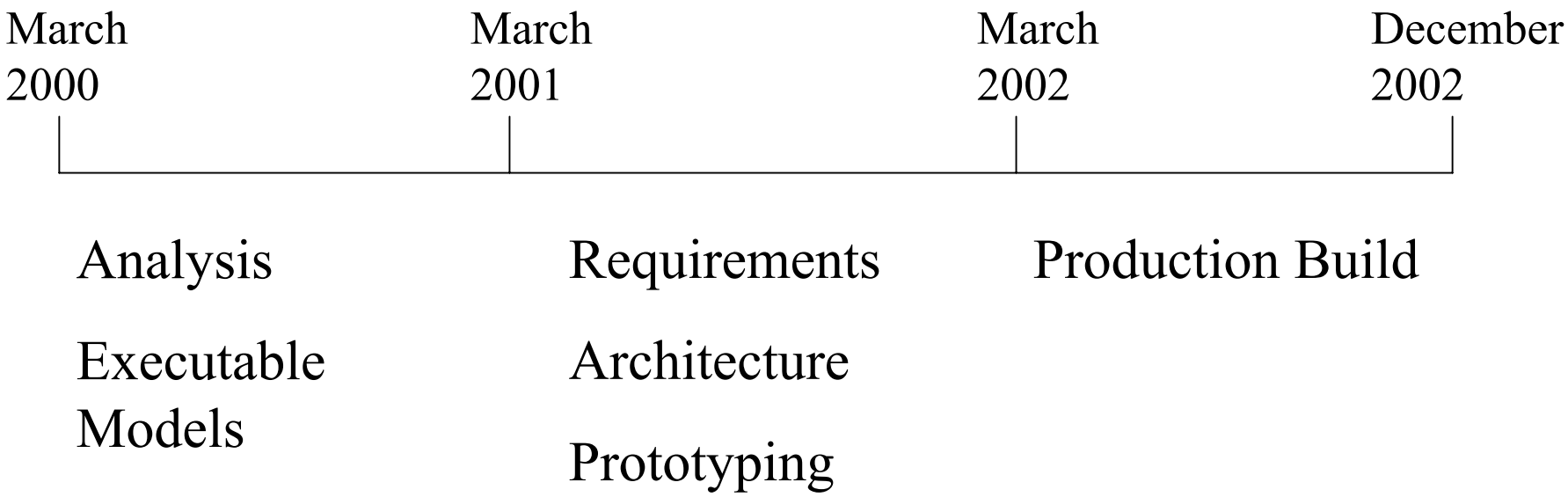
Brief History of the CDSA

- Created by Intel, then sponsored by the Open Group.
- Reference Implementation, Version 2.0
  - Release 3.0: March 2000, Intel, Microsoft Windows 98 & NT.
  - Release 3.12: Oct 2000, Bull Group, Linux port.
  - One-half million lines of C.
- Deployed:
  - MacOSX (Apple)
  - HP-UX (Hewlett-Packard, plug-ins from AT&T)
  - Tru64 (Compaq)

The CDSA Project at Galois Connections

Goals: Analyze and model the CDSA.

Build a high-confidence version of the CDSA.



Outline

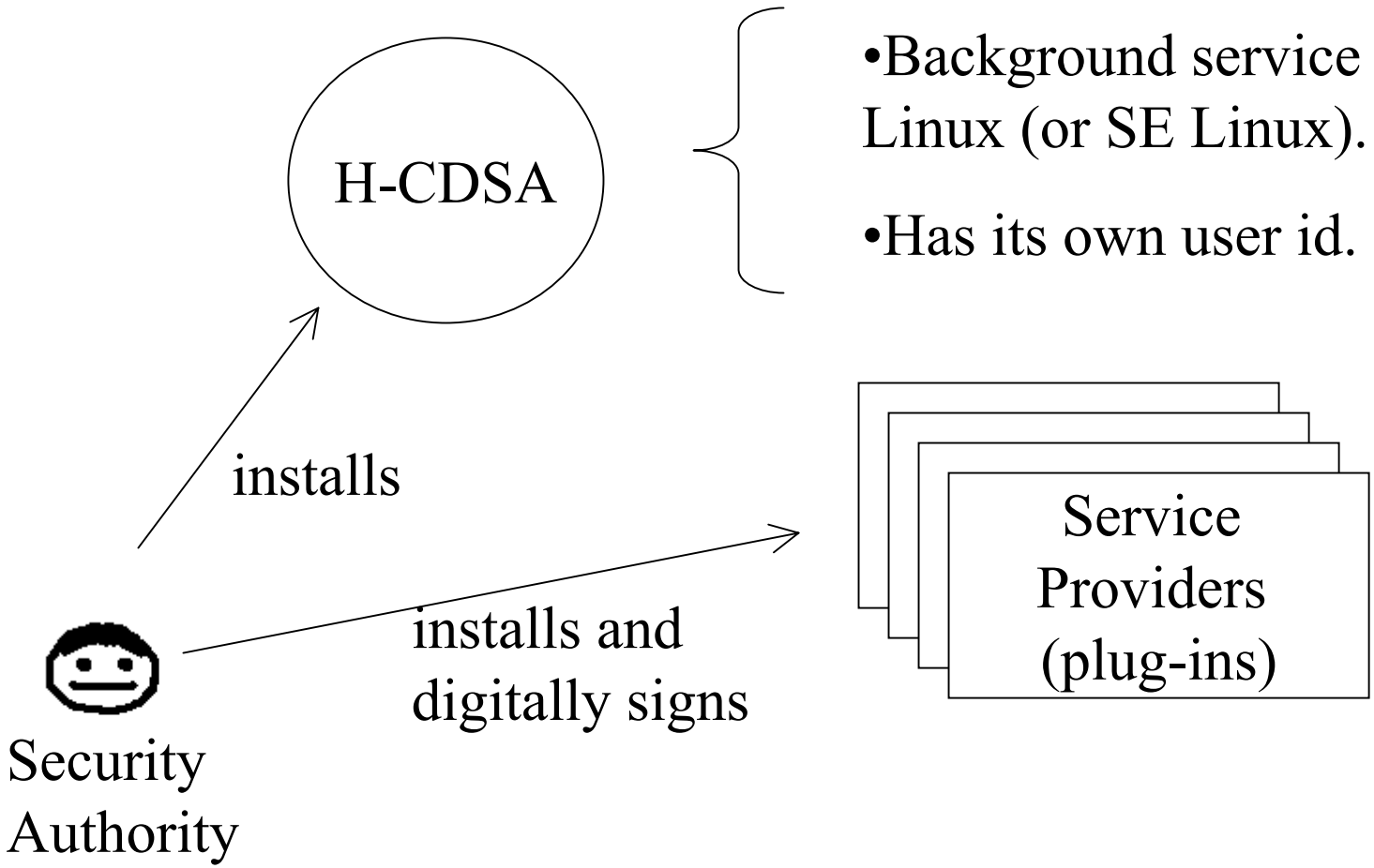
- What is the CDSA?
- **What is the H-CDSA?**
- Selected technical highlights:
  - Secure IPC on Linux
  - Remote Procedure Call
  - Trust Policies

The H-CDSA

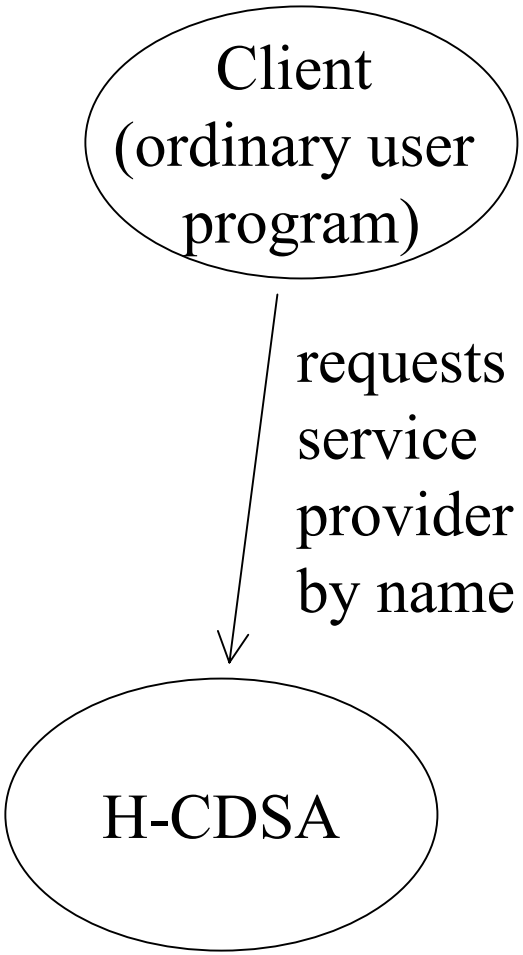
"High Confidence CDSA"

- Limited size of our project ➔ the H-CDSA must be much smaller than the CDSA.
- To reduce the size:
  - Re-engineer for simplicity.
  - Use functional programming technology.

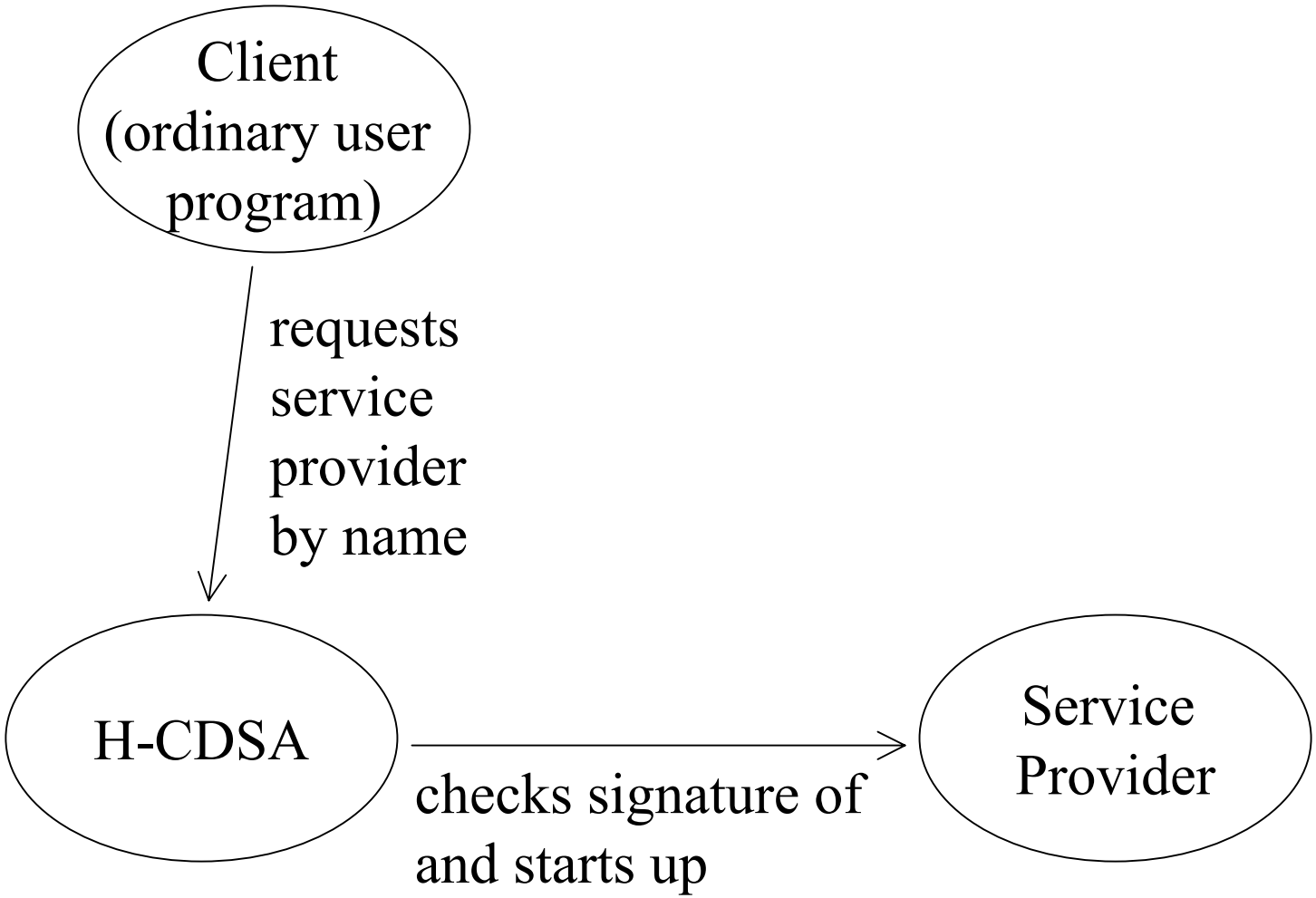
Installation of the H-CDSA



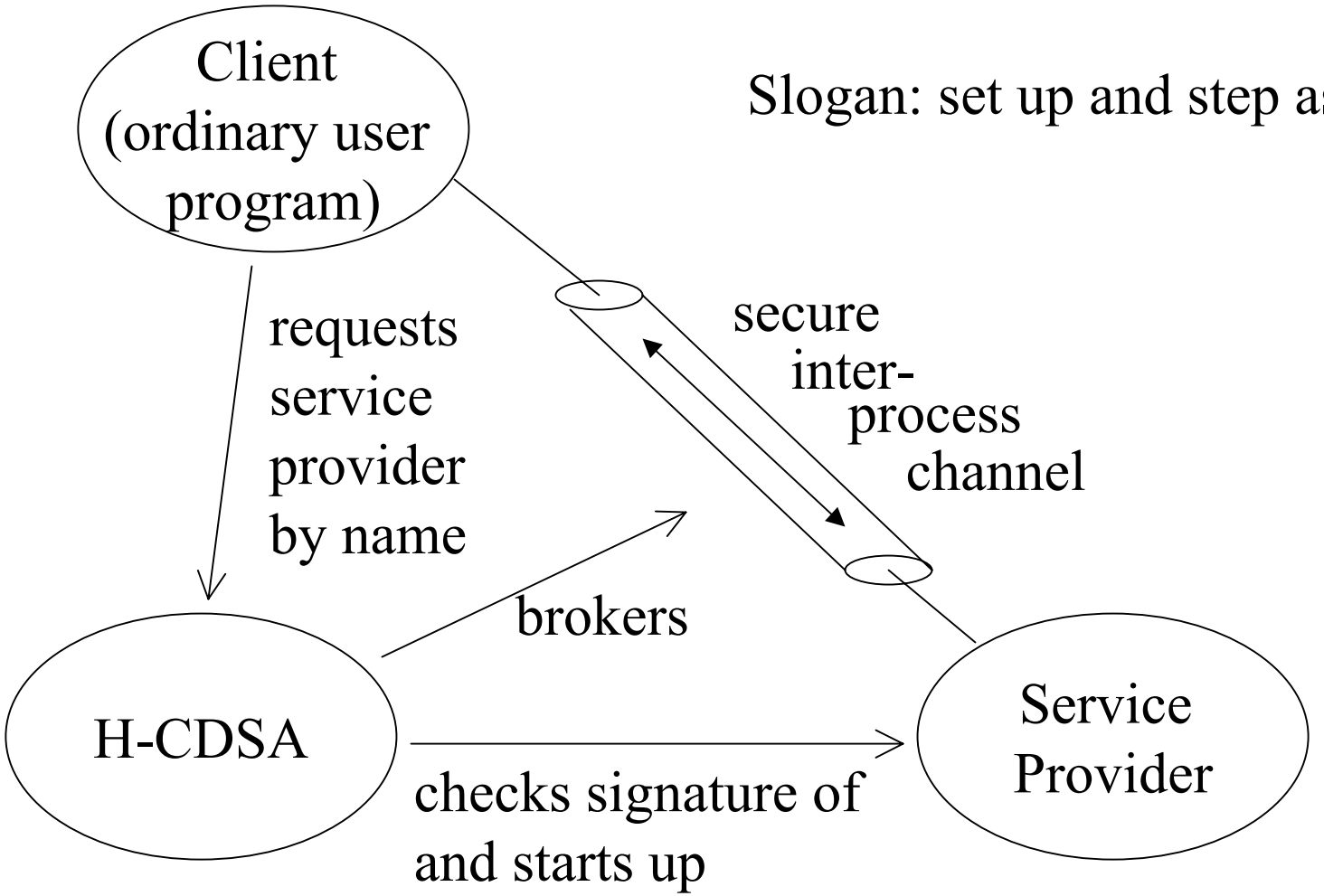
Obtaining Services via the H-CDSA (1)



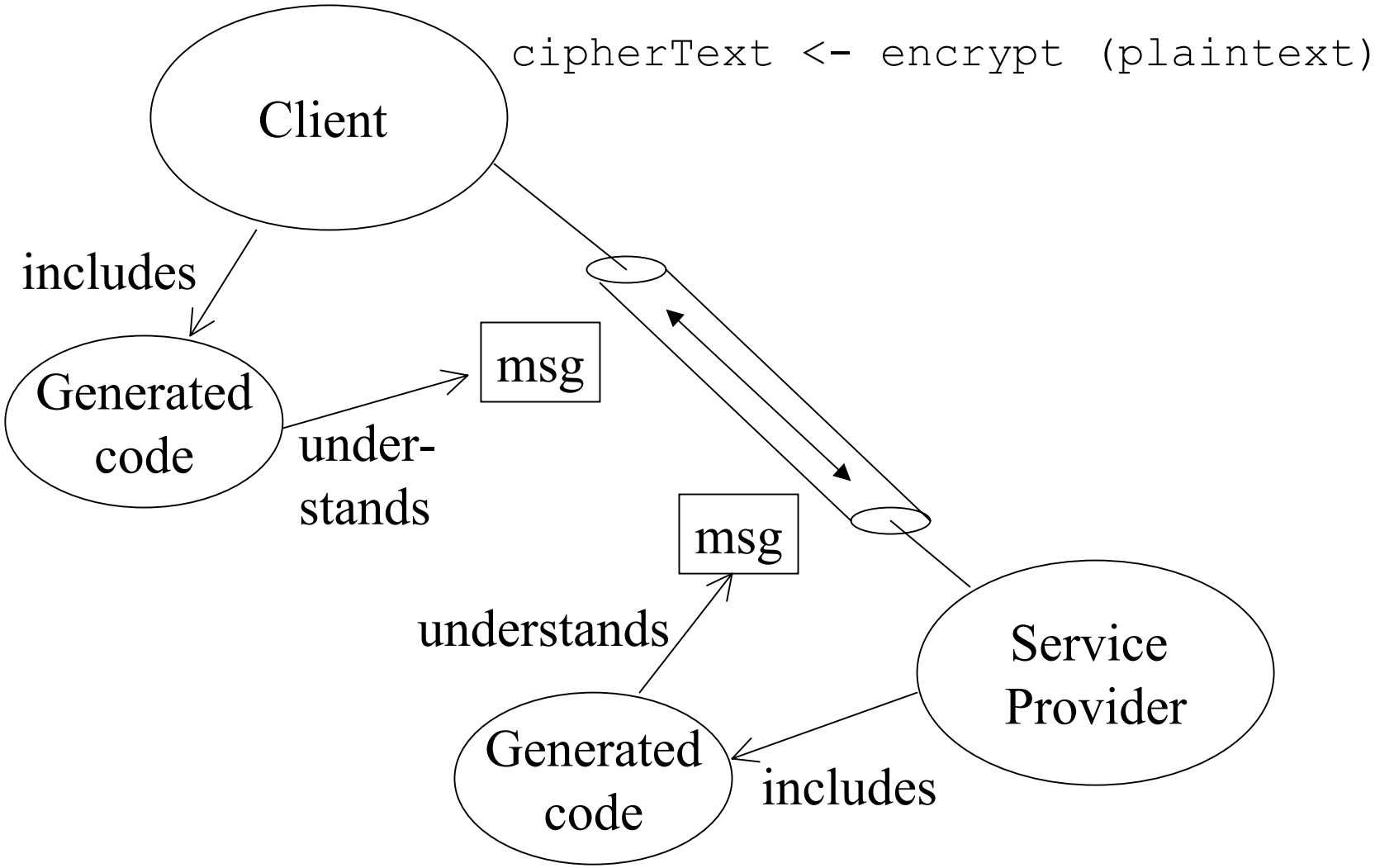
Obtaining Services via the H-CDSA (2)



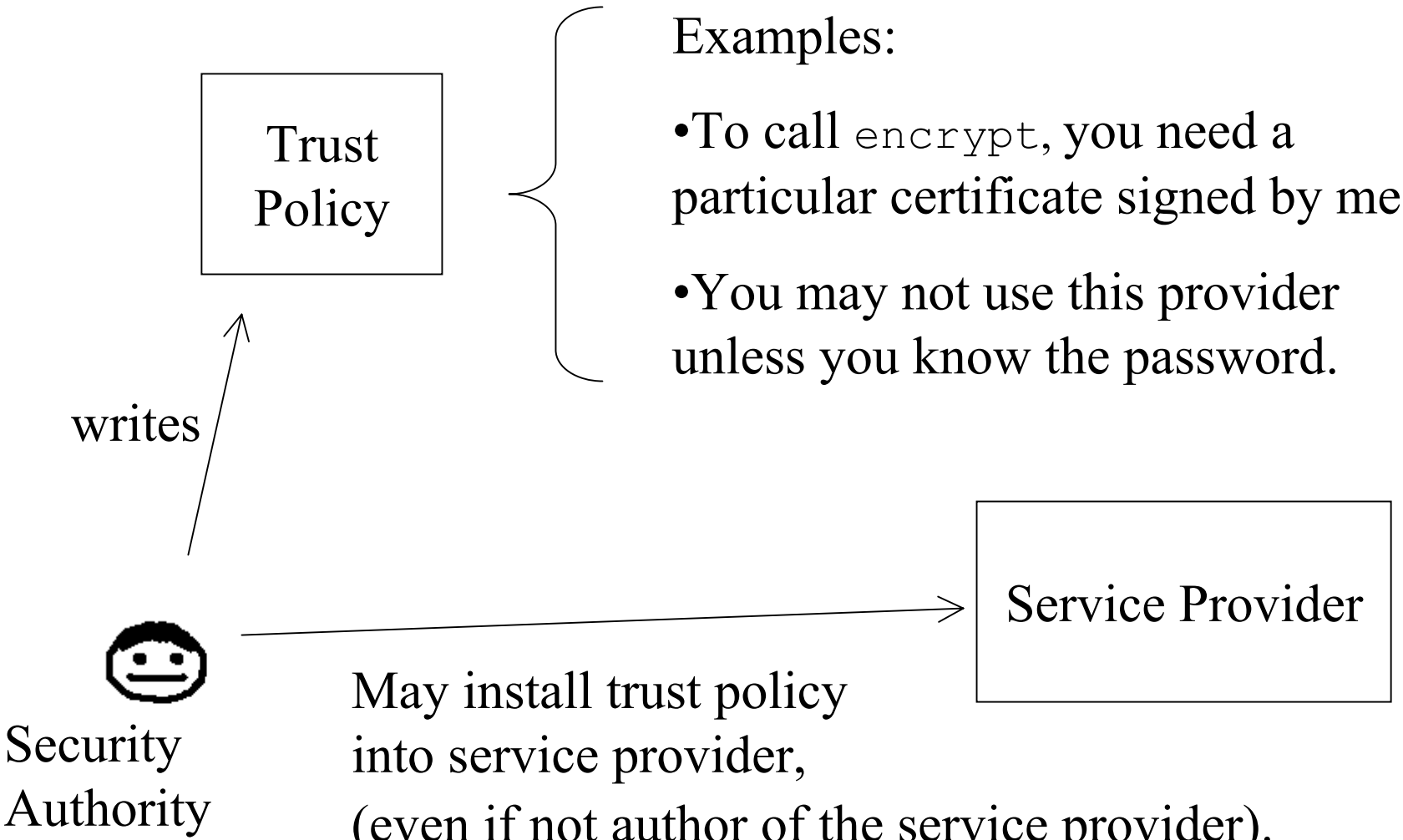
Obtaining Services via the H-CDSA (3)



Interaction Using Remote Procedure Calls



Trust Policies



Some Security Goals

What	Why
Client cannot tamper with H-CDSA or service providers.	H-CDSA and providers have different processes and user ids.
Client cannot bypass trust policy.	Trust policy enforced in provider.
Client knows service provider is authentic.	H-CDSA checks signature of provider before starting it.
Third party cannot eavesdrop.	OS keeps inter-process channel private.
Service provider cannot access client's memory.	Provider and client have different processes and user ids.
Separation between clients.	H-CDSA design goal.
Separation between service providers.	Provider confined to its own portion of file system.

CDSA vs. H-CDSA

(a partial list)

CDSA	H-CDSA
One-half million lines of C.	Goal: 20,000 or less lines of Haskell.
Client, CDSA, providers in same process. CDSA tries to provide isolation.	Separate processes and user ids. H-CDSA brokers connection, then steps aside.
Defines standard interfaces for providers.	Allows arbitrary interfaces for providers.
Specific format for access control lists, specifies where and how they are used.	Trust policy chosen by security authority at configuration time.

Formal Methods

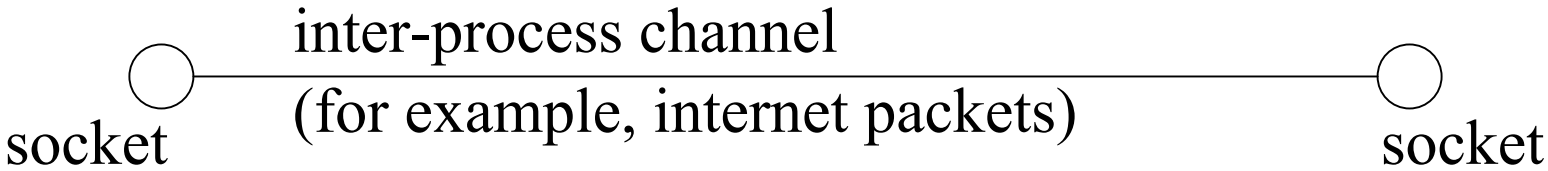
- Functional Programming in Haskell
  - Declarative, high level of abstraction.
  - Powerful type system.
  - Highly effective combination of pure mathematics and ability to run the program.
- Theorem Proving
  - Model a few small portions of the system.
  - Using Isabelle/HOL.

Outline

- What is the CDSA?
- What is the H-CDSA?
- **Selected technical highlights:**
  - Secure IPC on Linux
  - Remote Procedure Call
  - Trust Policies

Secure Inter-Process Communication on Linux

- H-CDSA uses “Unix-domain sockets”.
- Sockets in general:

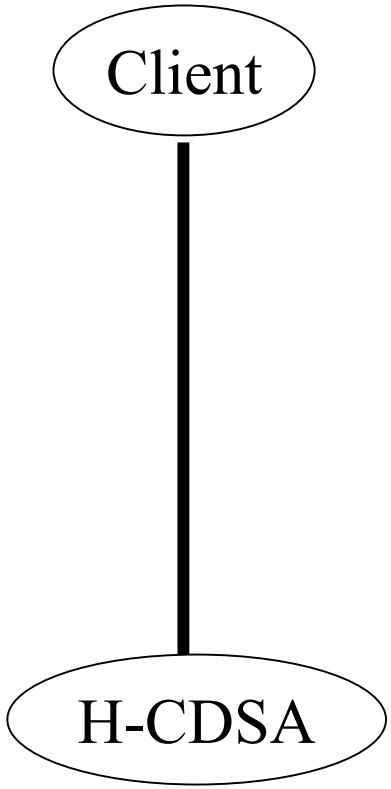


Processes use sockets much like ordinary files.
What is written into one end comes out at the other.

- Unix-domain socket:
  - The inter-process channel resides in the OS kernel.
  - Special ability to send a file descriptor (which is a connection to an open file) over the channel.

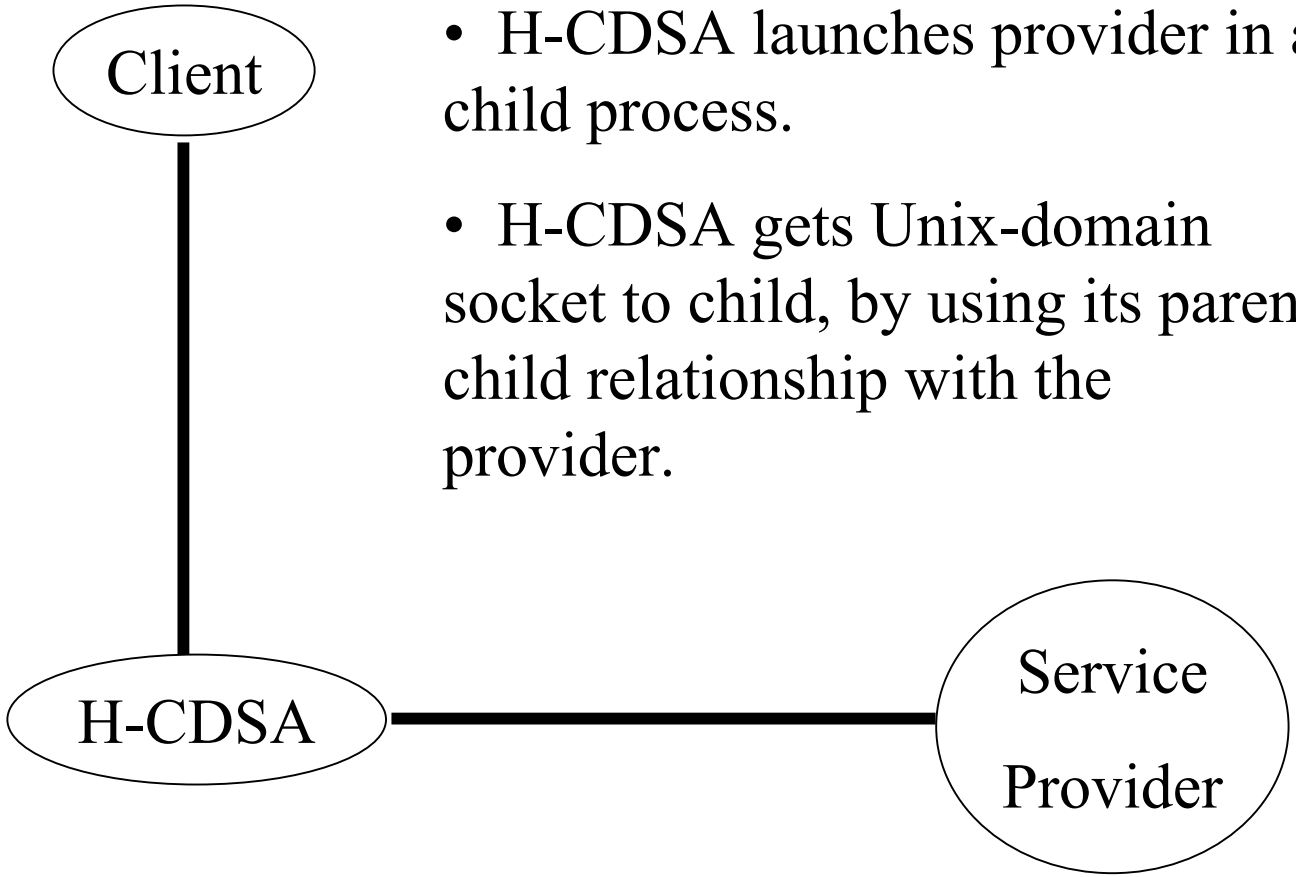
!

Brokering the Connection: Step 1



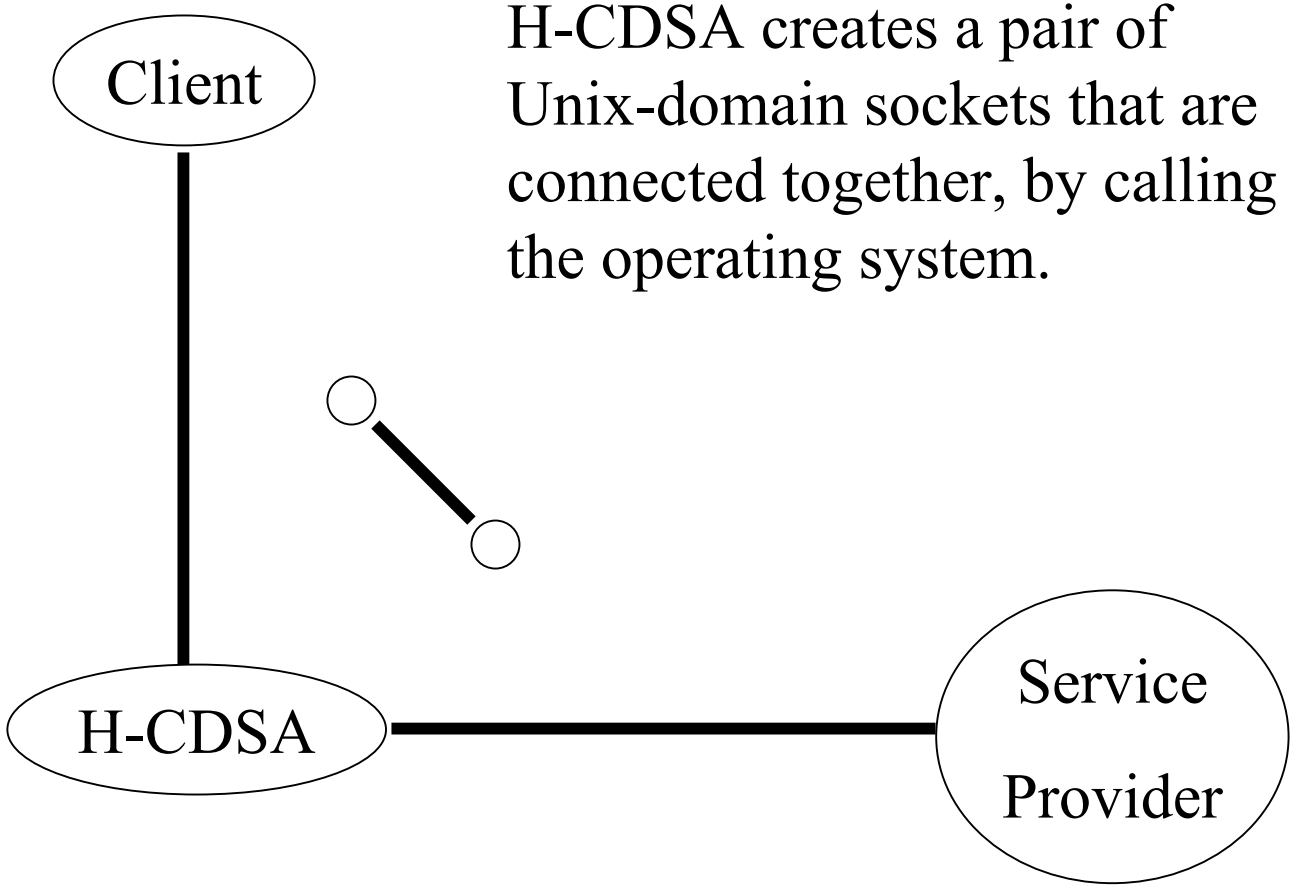
Client gets Unix-domain socket to H-CDSA, by knowing the name of an artifact in the file system.

Brokering the Connection: Step 2



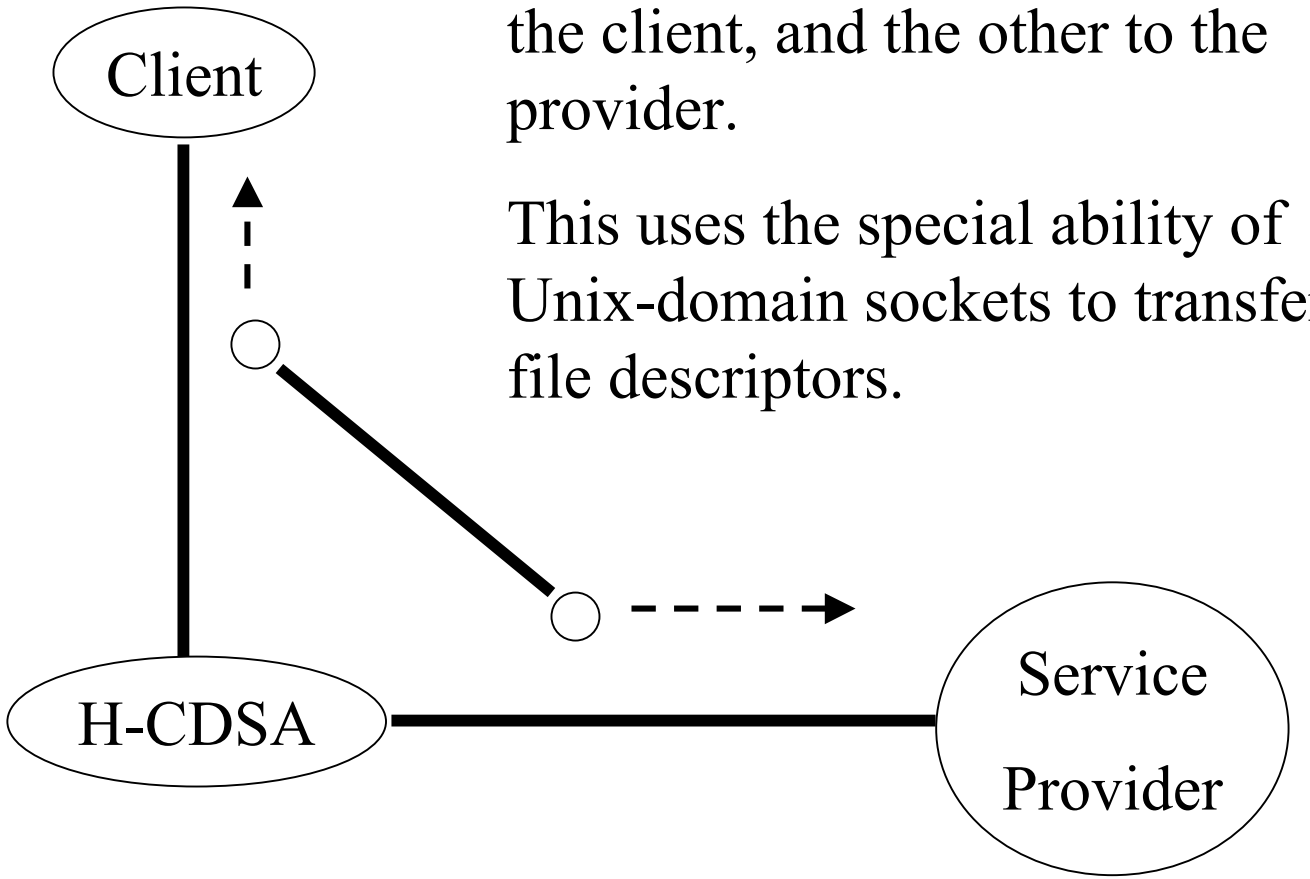
- H-CDSA launches provider in a child process.
- H-CDSA gets Unix-domain socket to child, by using its parent-child relationship with the provider.

Brokering the Connection: Step 3



H-CDSA creates a pair of Unix-domain sockets that are connected together, by calling the operating system.

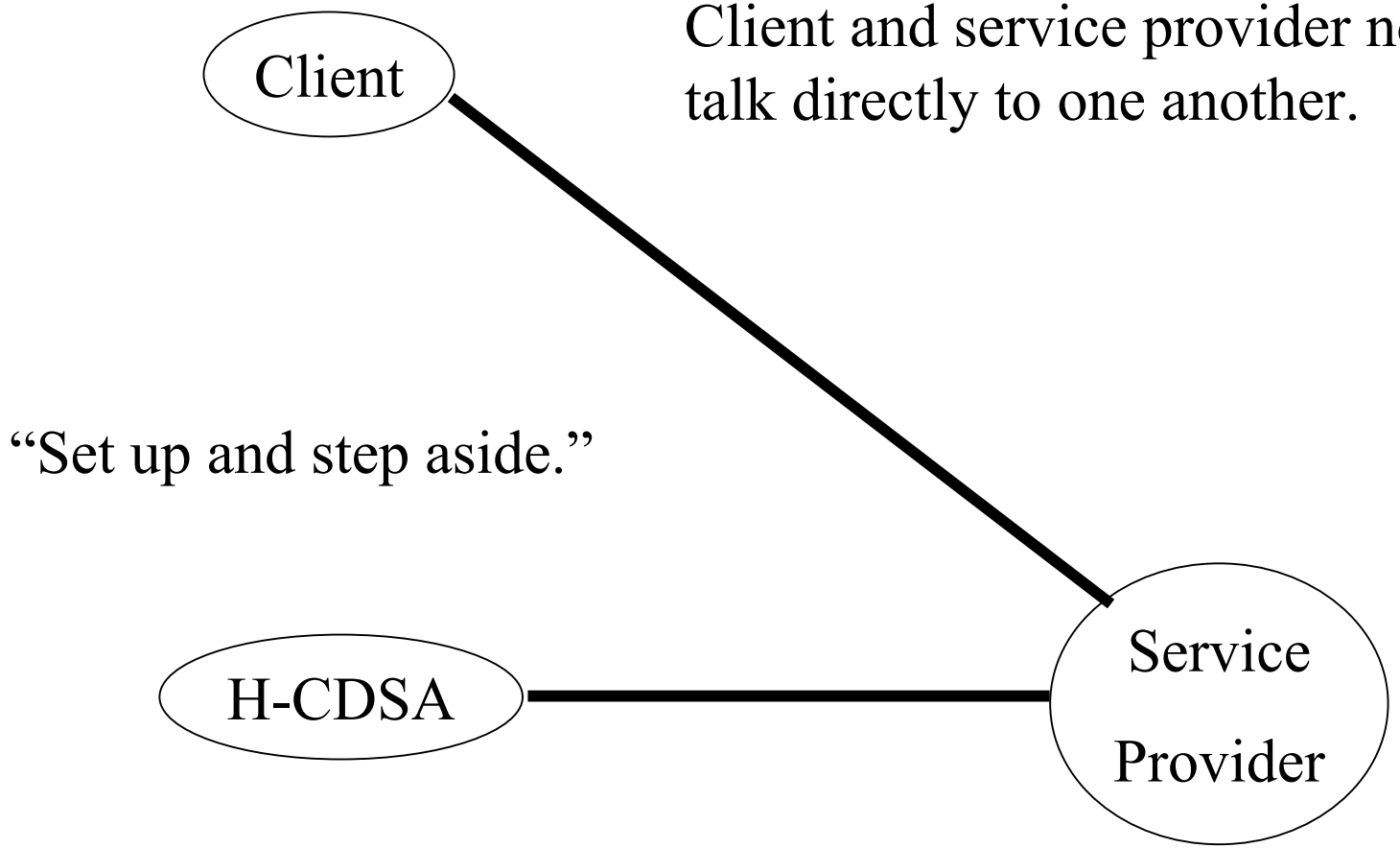
Brokering the Connection: Step 4



H-CDSA passes one socket to the client, and the other to the provider.

This uses the special ability of Unix-domain sockets to transfer file descriptors.

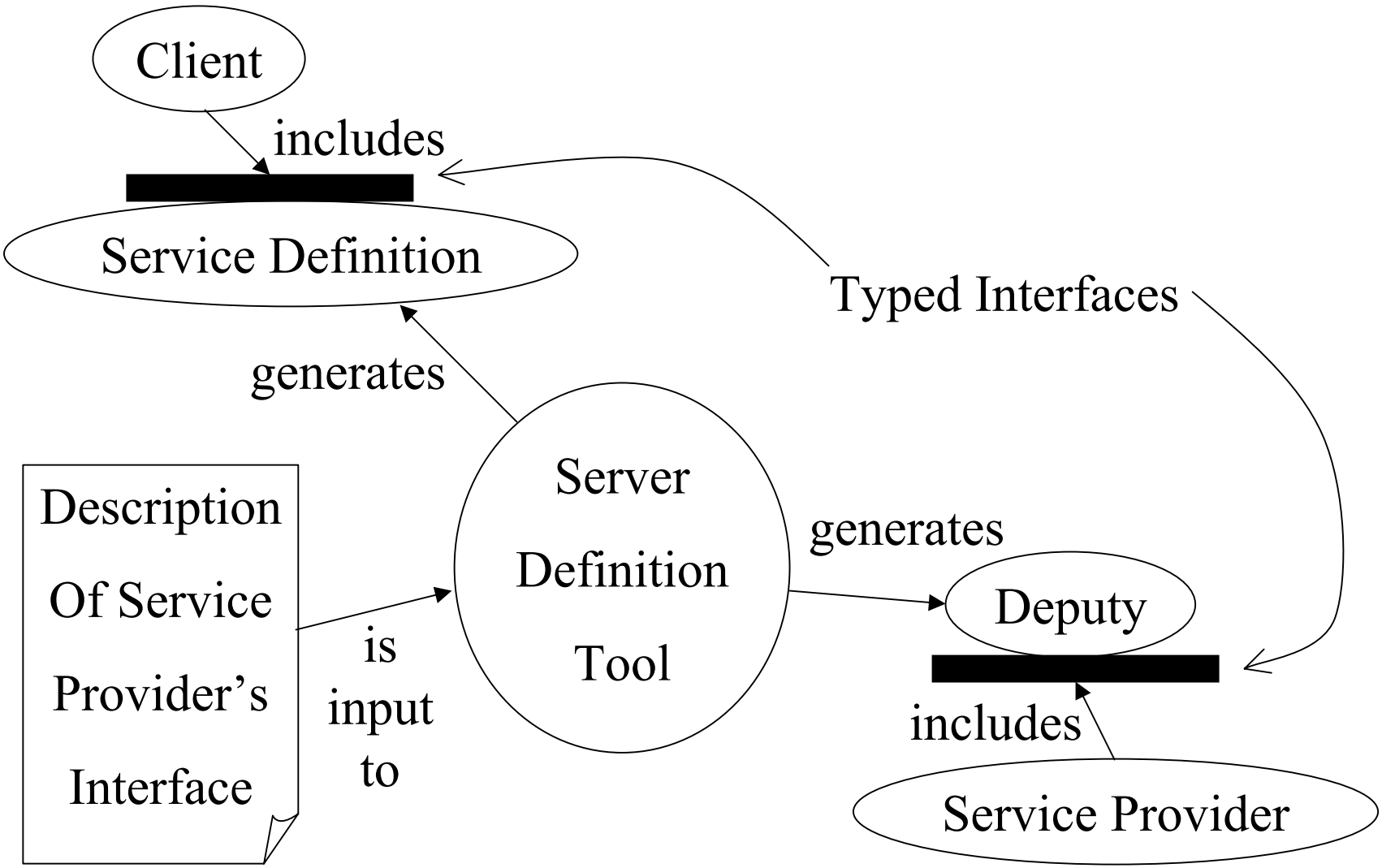
Brokering the Connection: Step 5



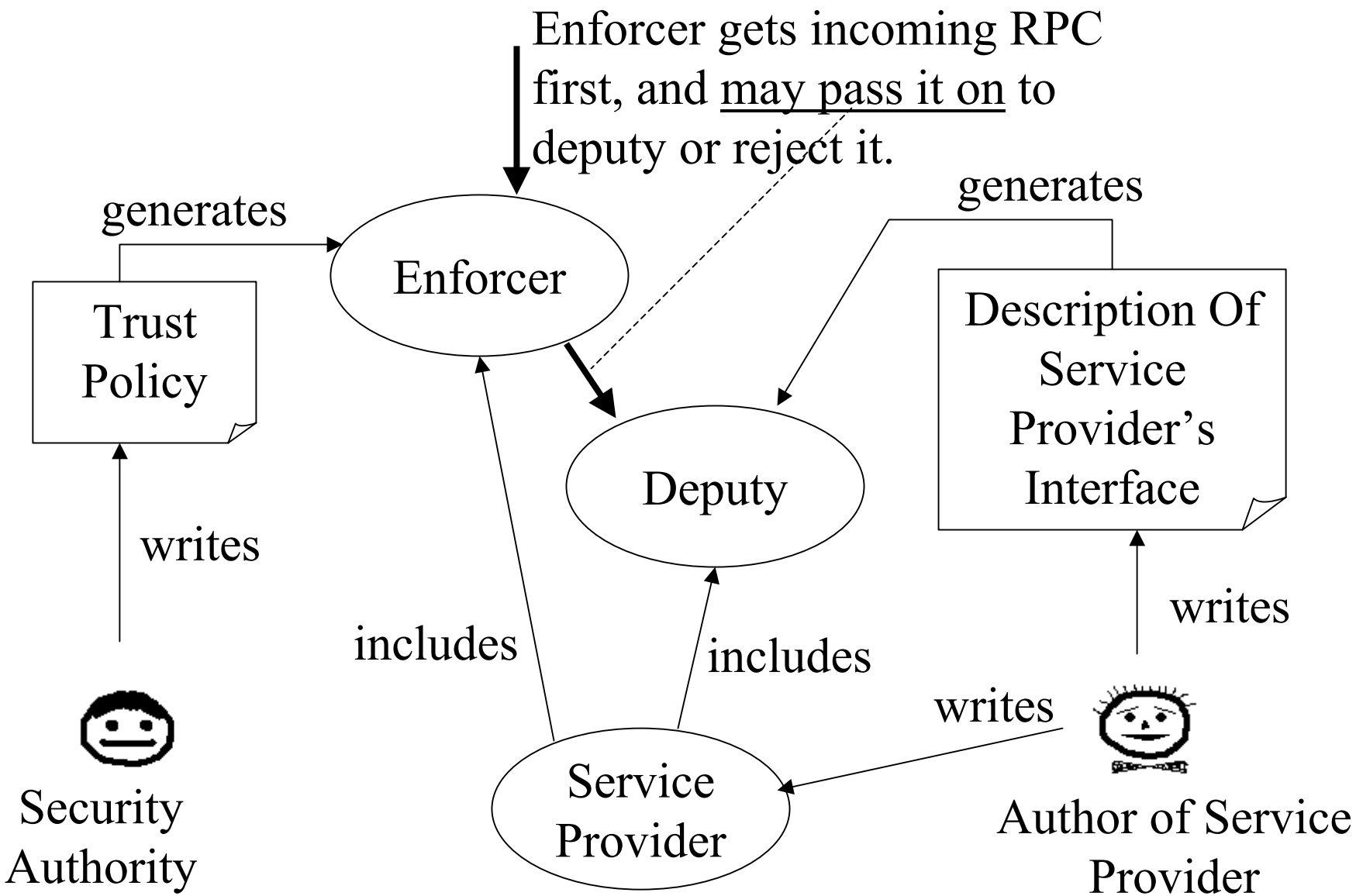
Remote Procedure Call

- Parameter types include:
  - Integer, float, Boolean, character string, block of 8-bit bytes.
  - Container types.
  - Procedure (such as a password callback).
  - Object (abstract datum with methods).
- Procedures as parameters work by passing tokens that name the procedures. (Objects are similar; they are like bundles of procedures.)

Remote Procedure Call



Trust Policies



Summary

- The CDSA is security middleware invented by Intel and standardized by the Open Group.
- The H-CDSA is a re-engineered, high-confidence version of the CDSA developed using functional programming.
- The H-CDSA brokers secure RPC with a “set up and step aside” philosophy, and permits the security authority to configure trust policies.
- The H-CDSA is scheduled for release by December 2002.