

Secure File System

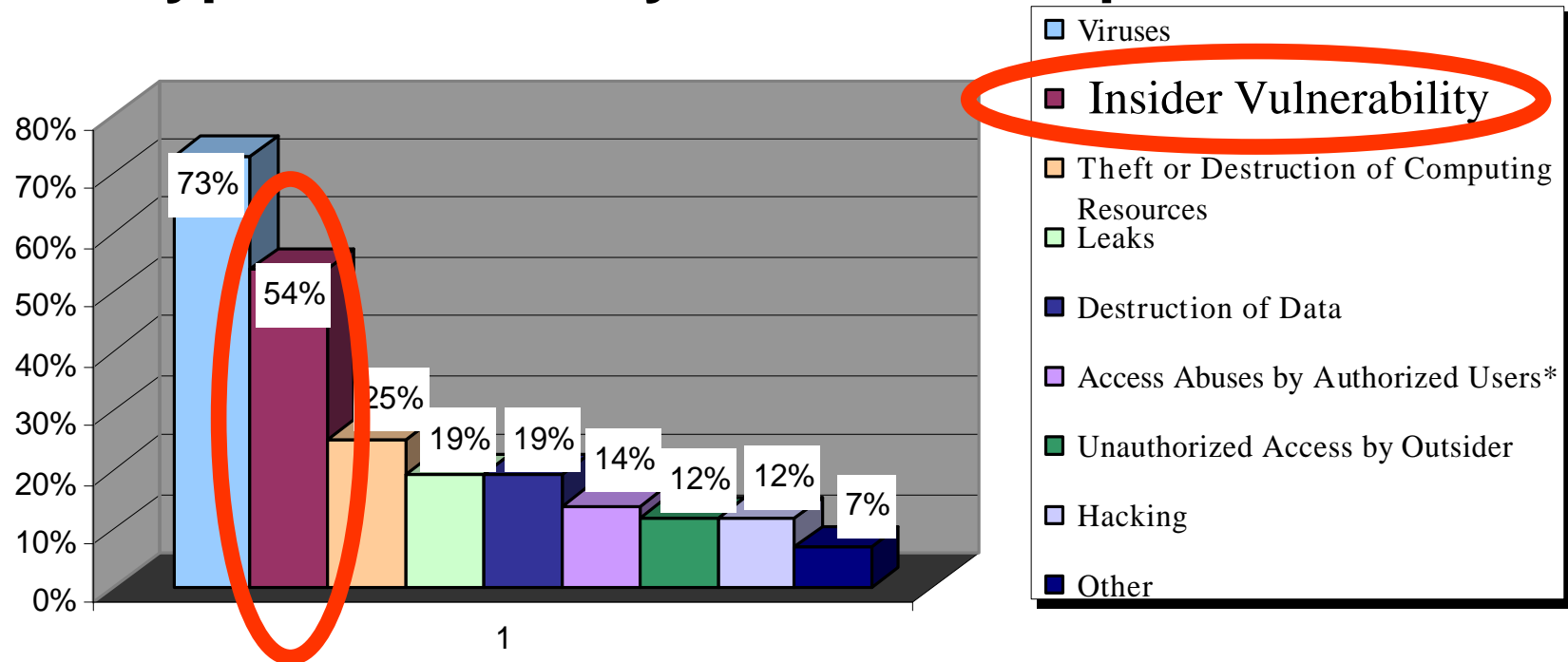
Jim Hughes,
Storage Technology Corporation

Chris Feist, Steve Hawkinson, Jeff Perrault, Matthew O'Keefe,
University of Minnesota

David Corcoran,
Purdue University

<mailto:SFS@SecureFileSystem.org>
<http://SecureFileSystem.org>

Types of Security Breaches Experienced



* *not including employees (e.g., business partner, vendor)*

31% of those suffering a breach said they suffered a business operations setback, while 18% said they suffered a financial loss.

Source: Information Security magazine June 1998 pp17.

(Fire)Walls are insufficient

- **Does not solve the hard problems**
 - Insider Threat
- **What is needed?**
 - harden all hosts
 - ◆ Every host is its own firewall?
 - Strong human authentication
 - ◆ Anonymity is seldom needed within a company
 - ◆ Non Repudable audit trails
 - ◆ Deterrence
 - “Information Security”
 - ◆ Secure File Storage mechanism

Insider Threat

- **Rogue Employees**
 - Every PC is a sniffer,
 - Hack from the inside
- **Information Systems (IS) organizations are large**
 - 5% of the typical company
 - hundreds to thousands of people
 - These people can access your data many ways
 - ◆ Email admin - surf email
 - ◆ Backup Admin - surf your backups
 - ◆ Network admin - sniff network
 - ◆ File Server admin - surf your files
- **Employees are necessary**

To build a Secure Information storage

■ Two Ways

- Secure the entire enterprise
 - ◆ Many times larger and more complex than securing a single system
 - ◆ Becomes harder as the organization gets larger
- End to End security
 - ◆ Data is protected from the producer to the consumer
 - ◆ Covers Networks, Backups, SANs, File Servers
 - ◆ Solves the insider threat

Strategy

- Solve the hard problem
 - Insider Attack
 - End to End security

- Easier in the long term than security everywhere
 - More effective
 - More transparent
 - lower cost!

Cryptographic Information Protection

- **Cryptographic File System, CFS, Matt Blaze**
 - Shared files require shared keys
- **Satan File System, SFS, CMU**
 - Interesting demonstration, Lib.c modifications
- **Distributed File System, DFS, IBM**
 - “Security is a network problem”
- **Networked Attached Secure Disks, NASD, CMU**
 - File system has all the master keys
- **Encrypted File System, EFS, Microsoft Windows 2000**
- **Secure File System, SFS, STK**

EFS (Microsoft Windows 2000)

- Encrypts between the server and the disk (not client)
 - Solves the DOS boot problem

The root of these security concerns is [...] Availability of tools that allow access to NTFS files from MS-DOS® and UNIX operating systems makes bypassing NTFS security even easier¹.

- Networks in the clear

[...] EFS only addresses encrypting data on disk. It does not encrypt data that is transferred over the network. Windows NT provides network protocols such as SSL/PCT to encrypt data access over the network¹.

- Backup persona(s) have access to all the data
 - Is still “Administrative Security”
 - Multiple personas each have access to all the data
 - ◆ No multi-person control of data escrow

¹ Microsoft, Encrypting File System for Windows NT Version 5.0. 1998

Information Security Middleware

■ What is Middleware

- What the OS expects the Application to do
- What the Application expects the OS to do
- OS Independent - Unix, Windows

■ Provides Information Security

- to information that is either leaving or entering this machine

■ Current Research

- Encrypted Tape
- Secure File System

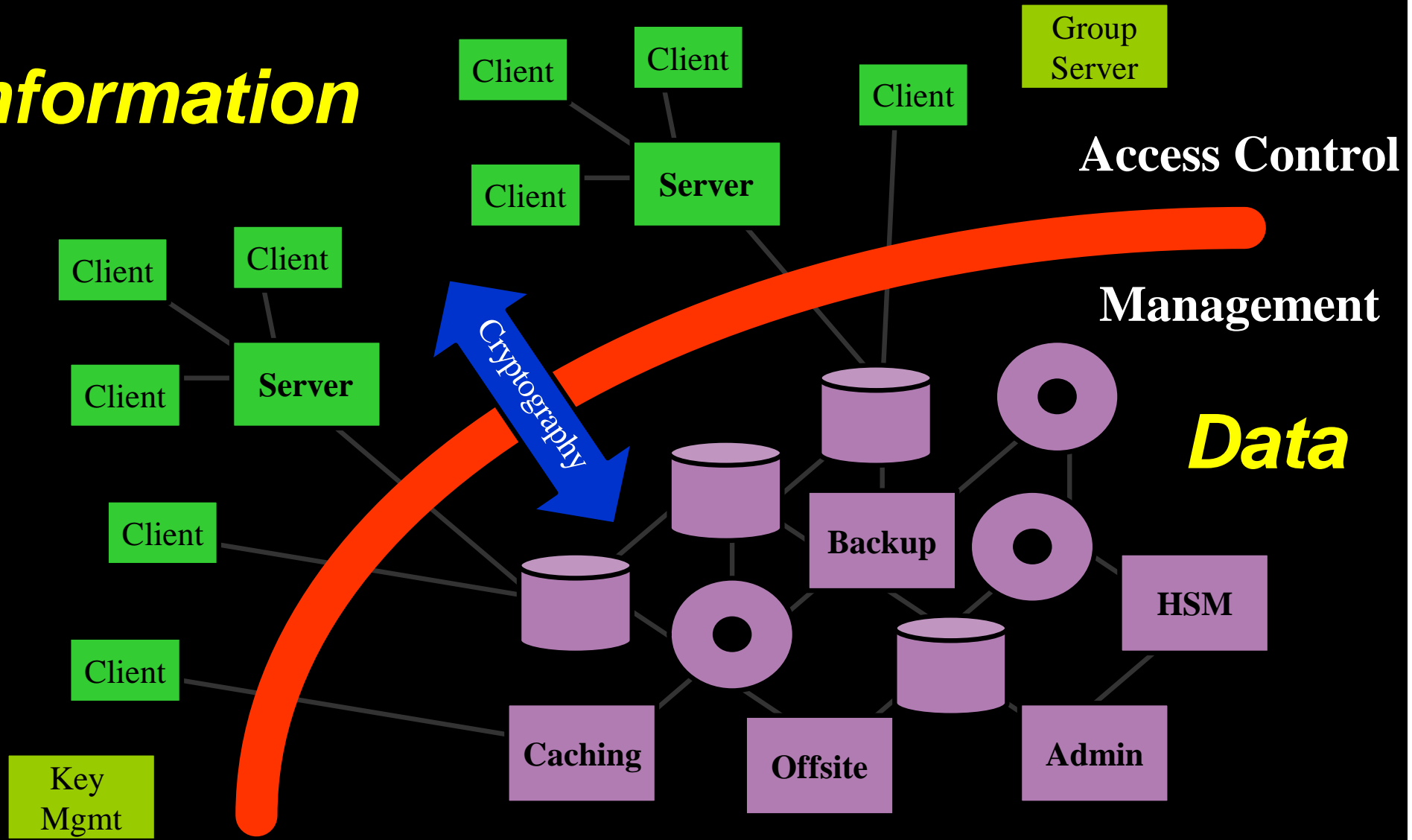
Separate information from data

- **Information** (Applications)
 - Understandable knowledge
 - Created for a purpose

- **Security Middleware** (OS Independent)
 - Access Control

- **Data** (File systems)
 - A managed set of bytes
 - Must not be lost
 - Mundane

Information



Data

Sharing

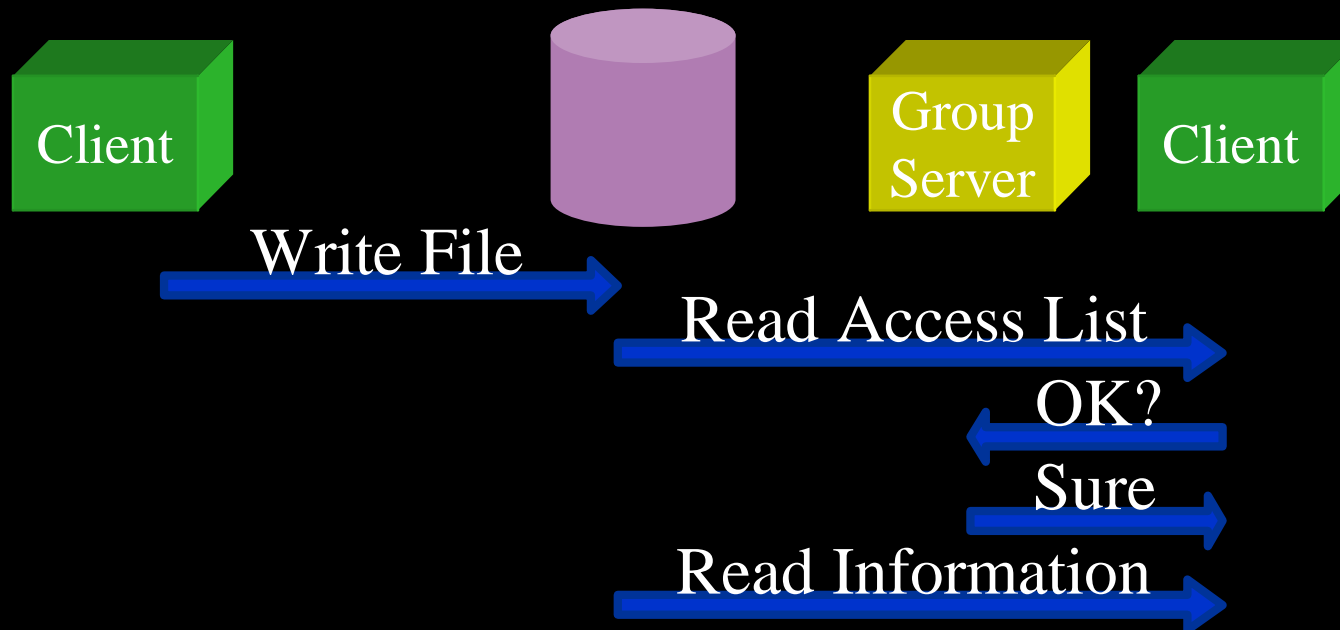
- Information to be shared has a purpose
 - Organizations have purpose
 - Encrypt information to organizations
 - ◆ Not to individuals
 - Organizations can be small or large
 - An individual is a group of one
- Let the organizations decide what to do with the information
 - It's their data
- Audit trails back to the owner

New Concept

- **RBAC based creation**
 - Information is protected to a consuming
 - ◆ organization
 - ◆ project
- **IBAC based consumption**
 - Information is unprotected to a member of an
 - ◆ organization
 - ◆ project
- **If you change organizations**
 - you lose your old access
 - you gain your new access
 - Stored information does not change

Group Server

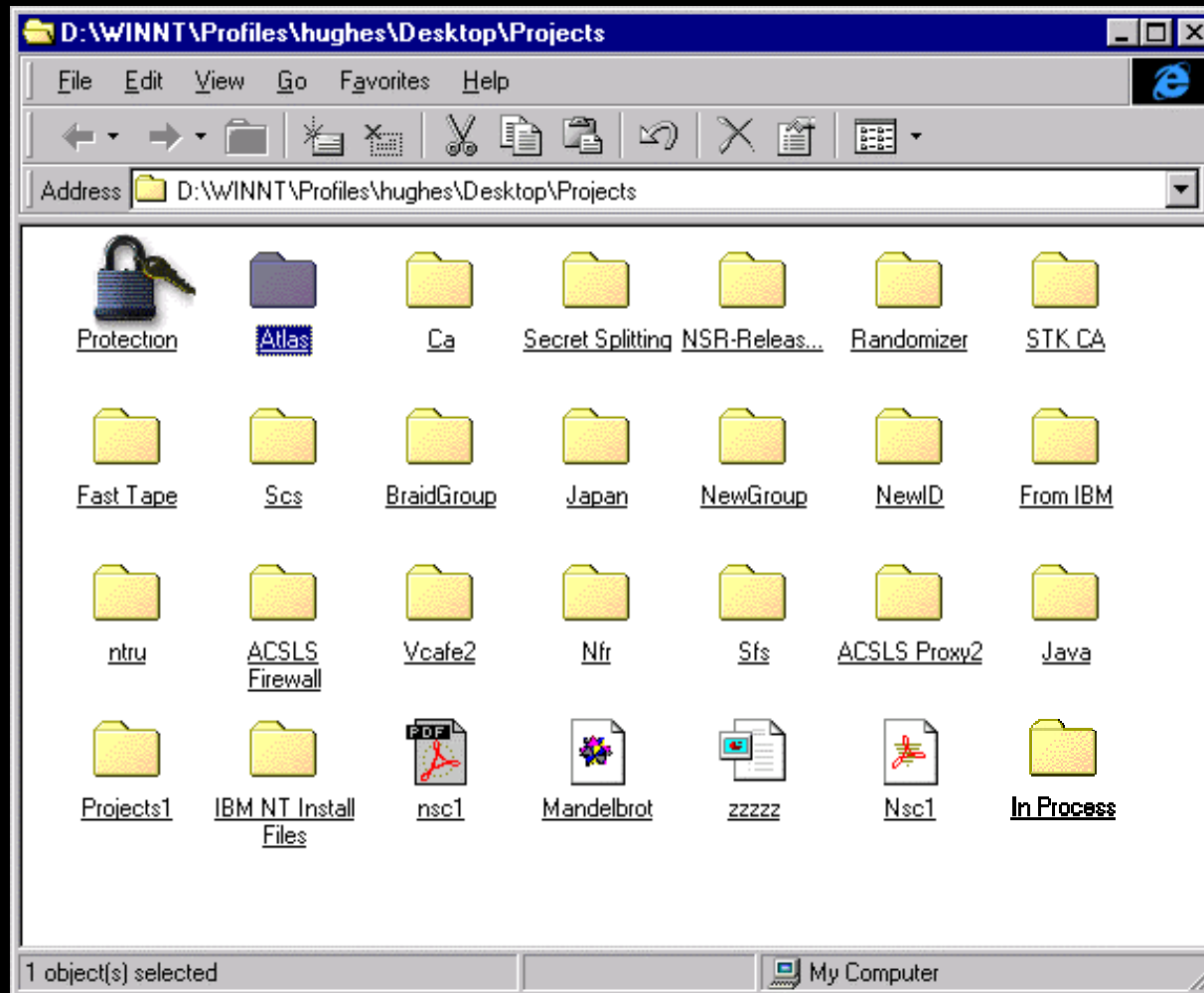
- Translates keys to individuals
- Trusted tamper resistant group key server
 - Reveals keys, does not decrypt files



Simple Human Interface

- Put a sharing “template” into each directory
- When files are written (dropped) into these folders, they are encrypted to the consumer(s)

Simple



Secure

- File intent is the user's responsibility
 - Non-reputable signatures
- Stored using impenetrable technology
 - Losing keys loses information
- Networks, backups, administrators, offsite storage are not a vulnerability

Roles/Definition

- **Producer**
 - Has (by definition) the data in the clear
 - Has the authority to define who can see their data
- **Consumer**
 - Needs the data in the clear (by definition)
 - Has (undeniable) ability to pass information on!
- **Group Agent**
 - Determines membership (need to know)
 - Subrogate information
- **Communications between Producer/Consumer**
 - Networks/Storage/Admins have no need to know the data

group (identifier(parameter-list_{opt}), $E_g(K)$)

- Responsible for
 - auditing of all accesses through this server
 - determining project or group membership
 - members revoked from the group
- Will disclose K if the caller fits into the *parameter-list*.
 - $E_g(K)$ translated to $E_{id}(K)$
- *parameter-list* can contain other group identifiers.
 - Recursively determine membership
 - Multiple audit points
 - ◆ different audit owners
- Virtual data enclave

any n of {consumer-list}

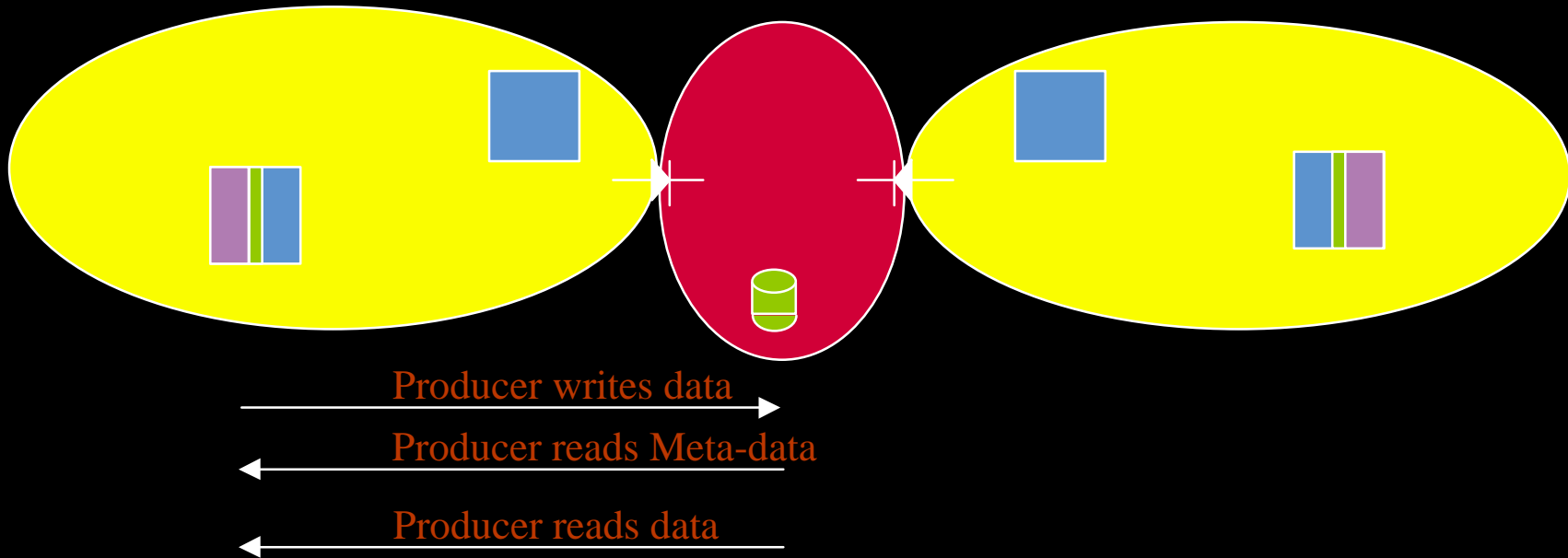
- list of m consumers each given $1/n$ th the key
- creates m linear equations of n variables.
- Each is linearly independent
- Designed such that having $n-1$ pieces provides no advantage

- True multi-person control of data recovery
 - Allows secure self escrow

Example - my private data

```
Purpose (  
  ident(me, Kme);  
)
```

Example - my private data



Example - my private data, with escrow

Purpose (any 1 of (
 $\text{ident}(\text{me}, E_{\text{me}}(K));$
 any 2 of (
 $\text{ident}(\text{ke1}, E_{\text{ke1}}(K_1));$
 $\text{ident}(\text{ke2}, E_{\text{ke2}}(K_2));$
 $\text{ident}(\text{ke3}, E_{\text{ke3}}(K_3));$
)
))

Macros

#Define escrow

any 2 of (

ident(ke1, $E_{ke1}(K_1)$);

ident(ke2, $E_{ke2}(K_2)$);

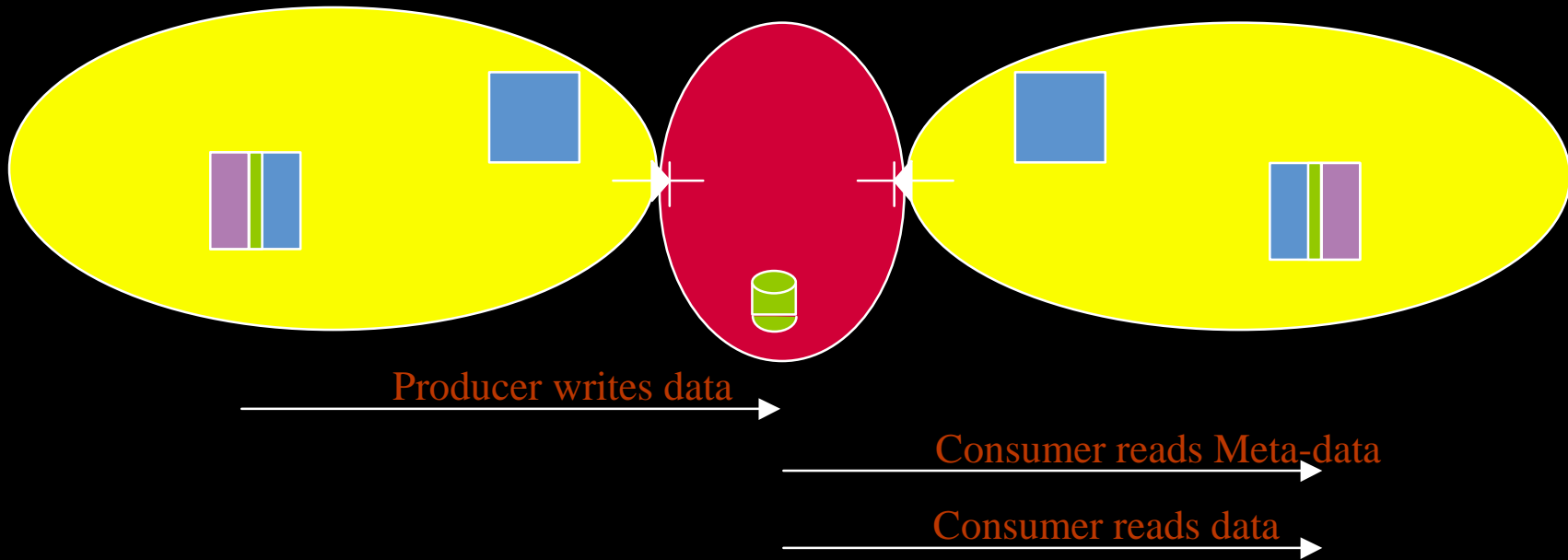
ident(ke3, $E_{ke3}(K_3)$);

)

Example - my data to you, with escrow

```
Purpose (any 1 of (  
  ident(you,  $E_{\text{you}}(K)$ );  
  escrow;  
))
```

Example - my data to you



Example - my data, you audit

Purpose (any 1 of (

escrow;

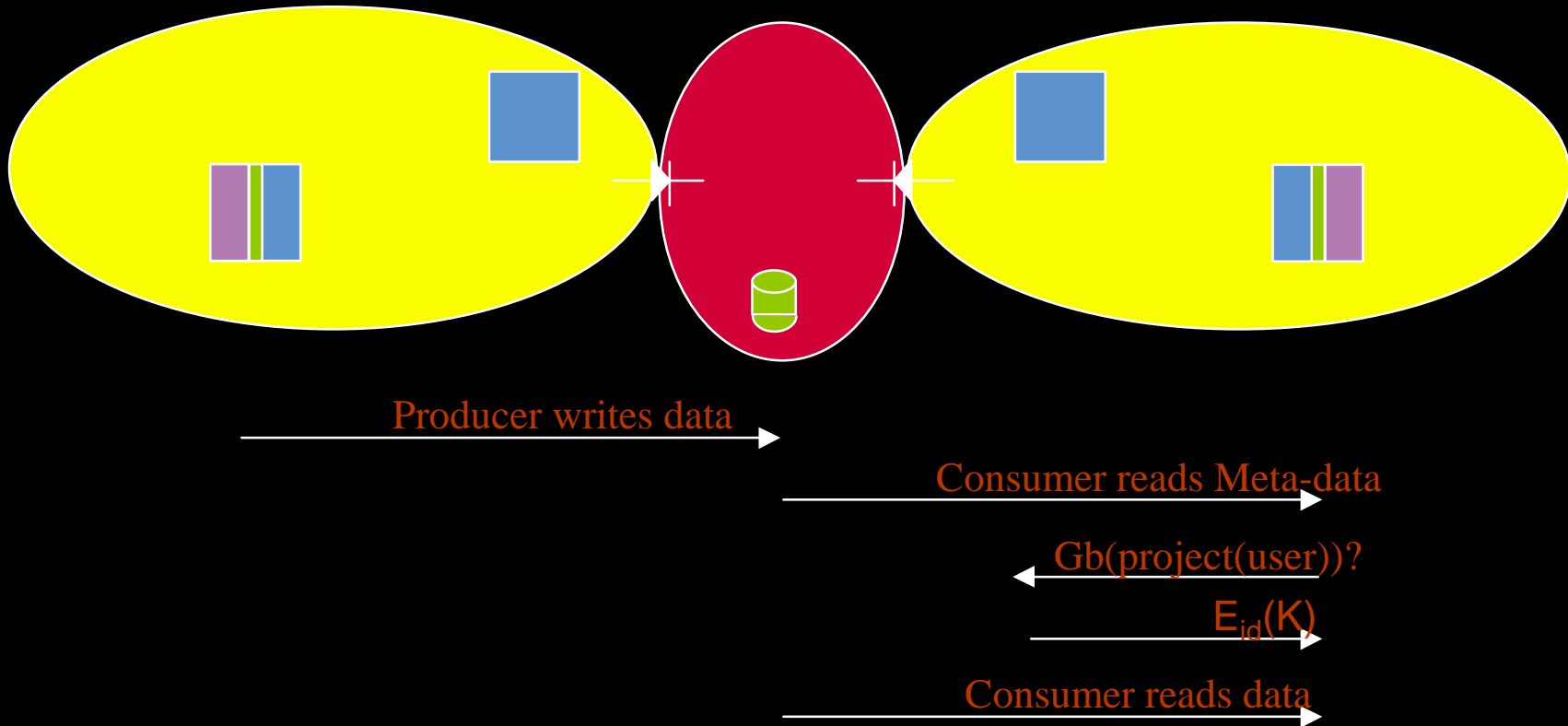
group(Gb(project_b), $E_{gb}(K)$);

group(Gc(project_c), $E_{gc}(K)$);

group(Gd(project_d), $E_{gd}(K)$);

))

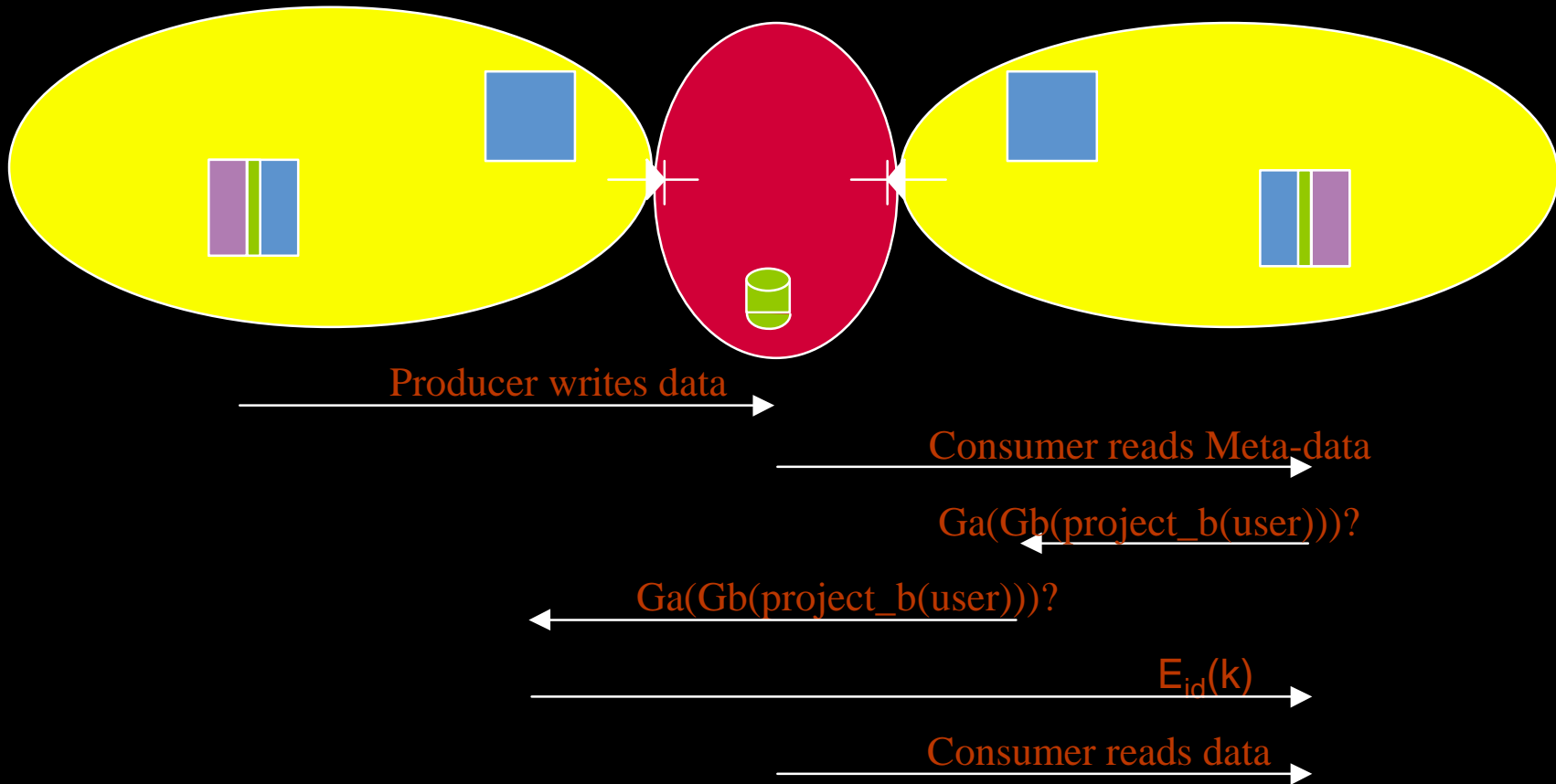
Example - my data, you audit



Dist project, group audit

```
Purpose (any 1 of (  
  escrow;  
  ident(me, Eme(K));  
  group(Ga(  
    Gb(project_b), Gc(project_c), Gd(project_d)  
  ), Ega(K));  
))
```

Dist project, group audit



Dist project, complete audit

Purpose (any 1 of (

escrow;

ident(me, $E_{me}(K)$);

any 2 of (

group($G_a(G_b(\text{project_b}), G_c(\text{project_c})), E_{ga}(K_1)$);

any 1 of (

group($G_b(\text{project_b}), E_{gb}(K_2)$);

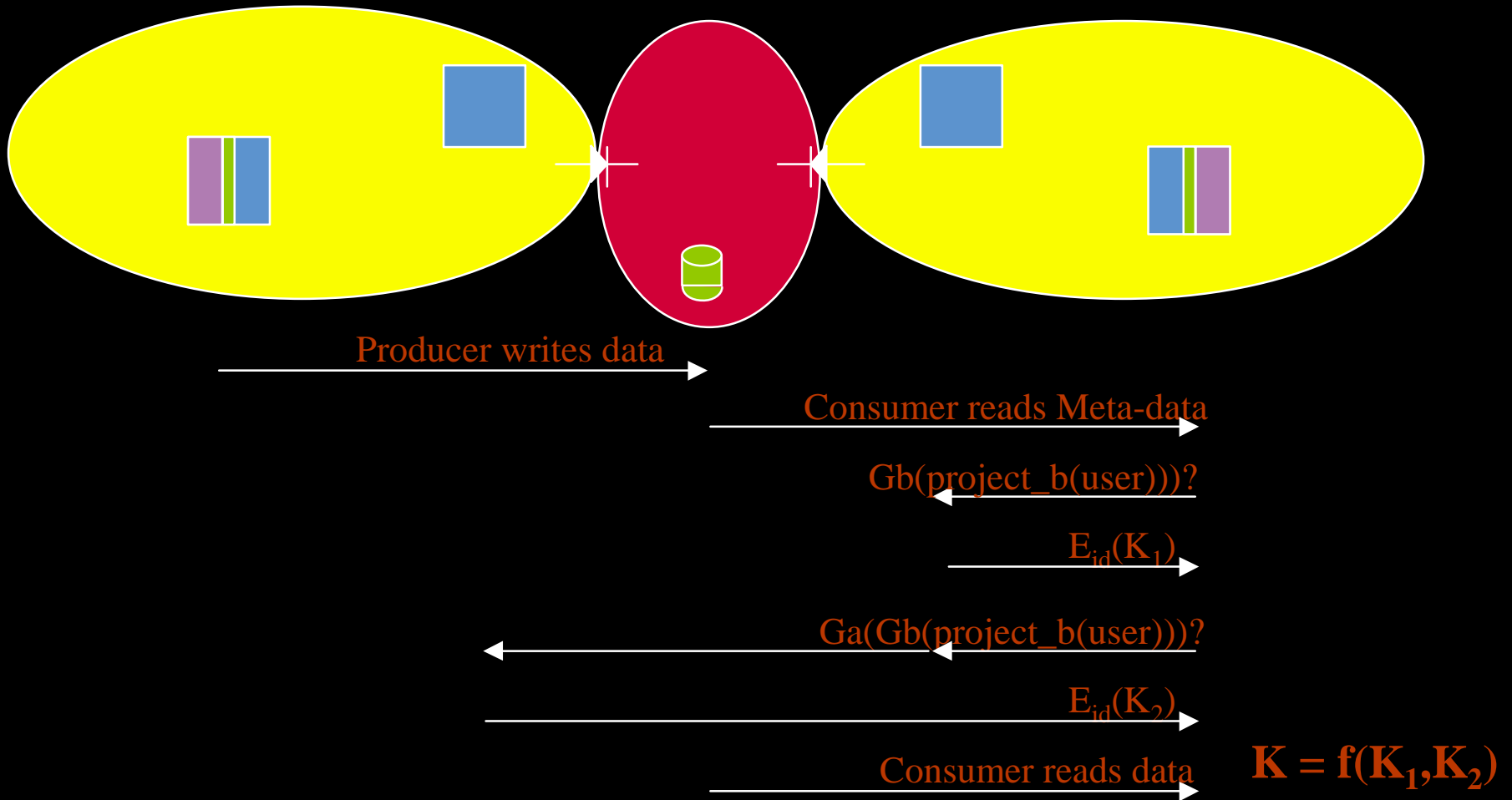
group($G_c(\text{project_c}), E_{gc}(K_2)$);

)

)

))

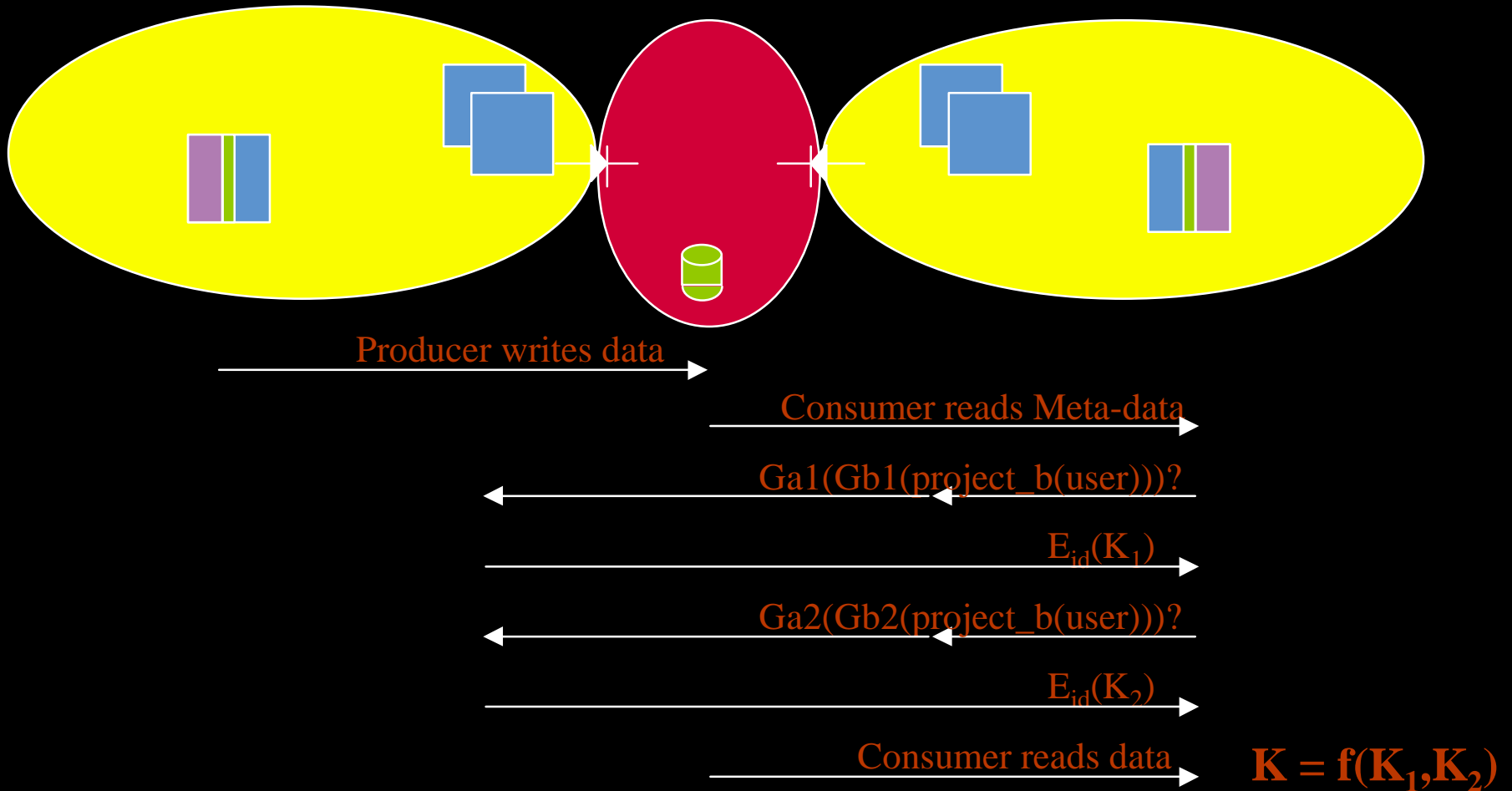
Dist project, complete audit



Dist project, centralized group audit

```
Purpose ( any 1 of ( escrow; ident(me, Eme(K));  
  any 2 of (  
    group(Ga1(Gb1(project_b)), Ega(K1));  
    group(Ga2(Gb2(project_b)), Ega(K2));  
  )  
))
```

Dist project, centralized group audit



Flexible security

- Defined by the producer
 - Data owner
- Protection levels
 0. Clear (today)
 1. Clear, non-repudiatable data
 2. Encrypted, consumer's group access/audit
 3. Encrypted, producer's group access/audit
 4. Encrypted, Watermarked copies

UofMN Program Focus

- Determine Mechanics
 - UFO
 - WrapFS
 - Group Server
- Transparent Operation
 - Simple User Interface
 - Smart Card Authentication (Java card)
- Simple Membership
 - Easy Administration
 - Secure Audit
- Source will be available

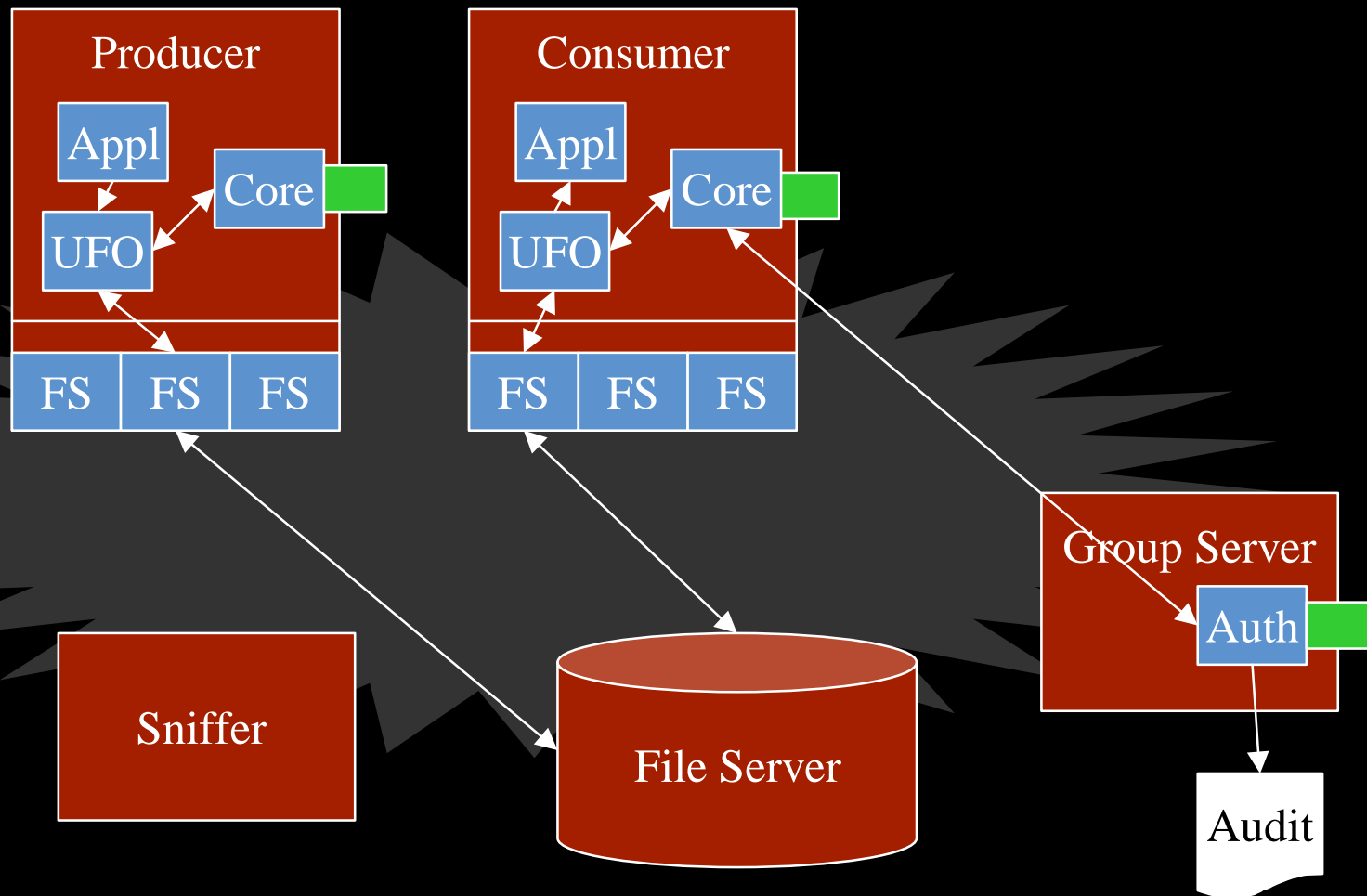
STK Focus

- **Cryptographic Definitions**
 - Protocol Definitions
 - ◆ **Public/Private/Hash/Signatures**
 - Tool Box definition (X.509), PKCS#11
- **Assured**
 - Software
 - Separate box
- **High Performance**
 - Hardware
- **Product**
 - Shrink Wrapped

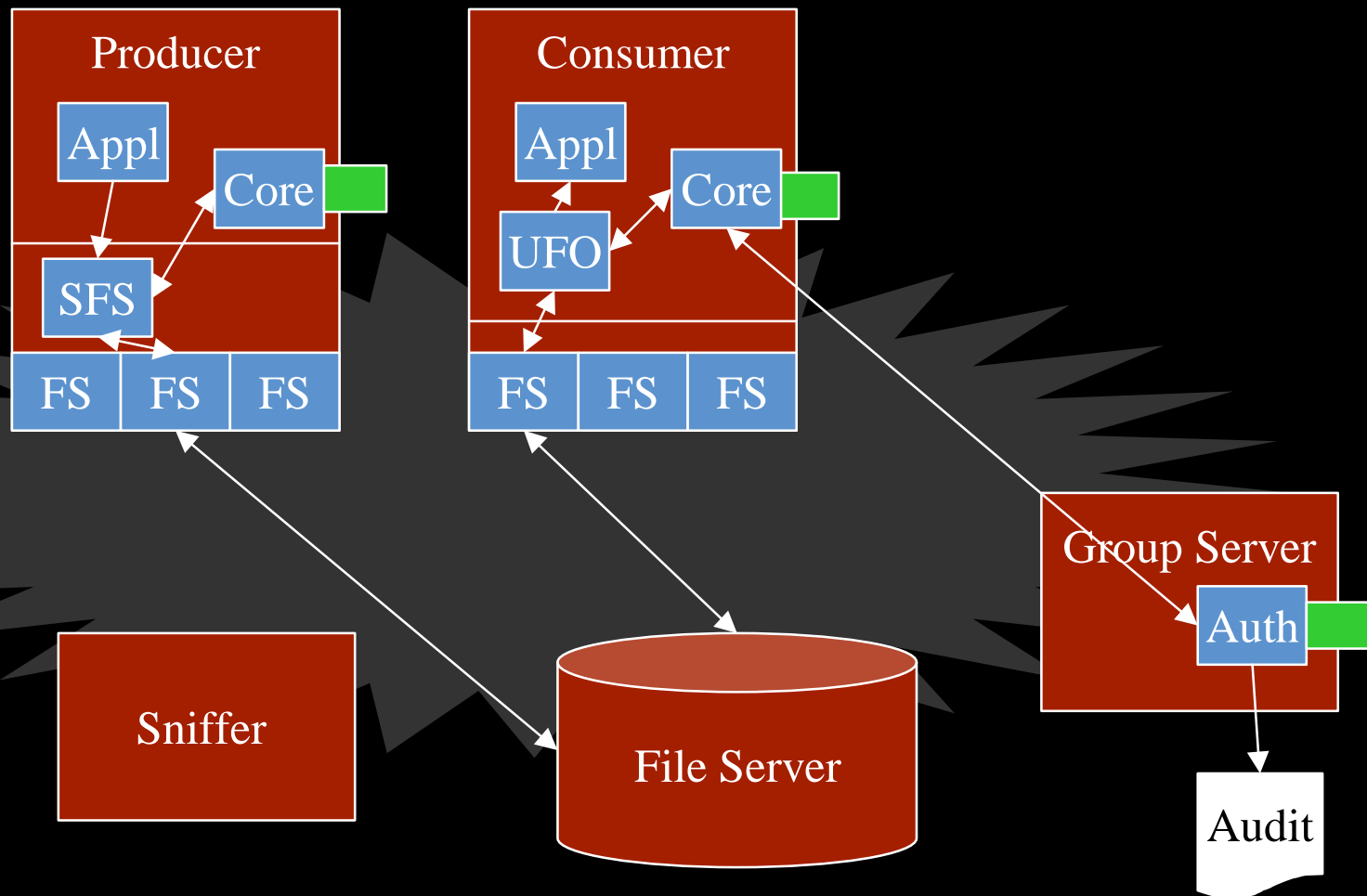
Program

- is not Trying to Windows Secure
 - Can not make a silk purse out of a sow's ear
 - but it will help
- may be useful for Secure Linux?
 - This program can provide trusted labels to and from storage

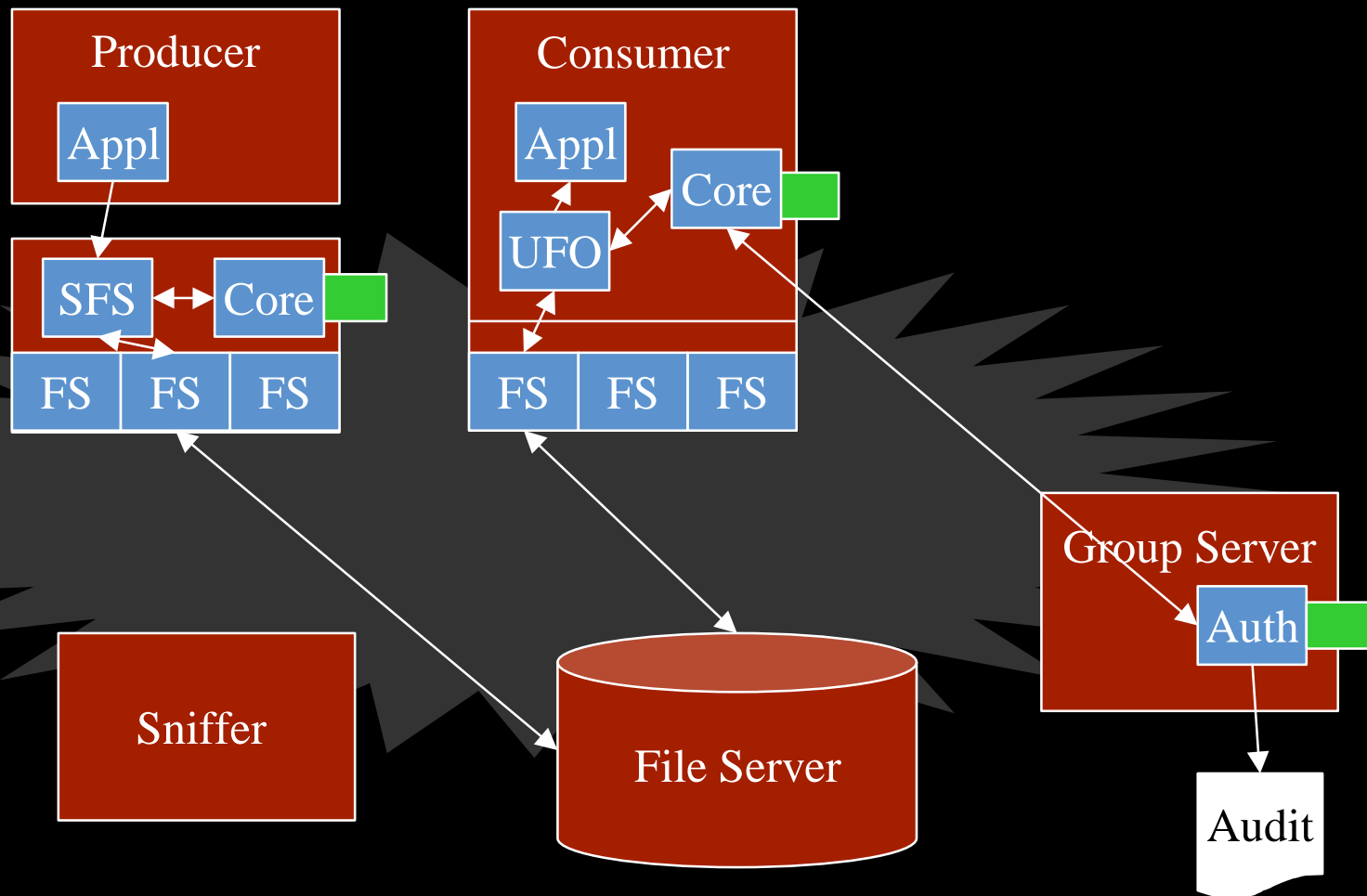
Demonstration Configuration



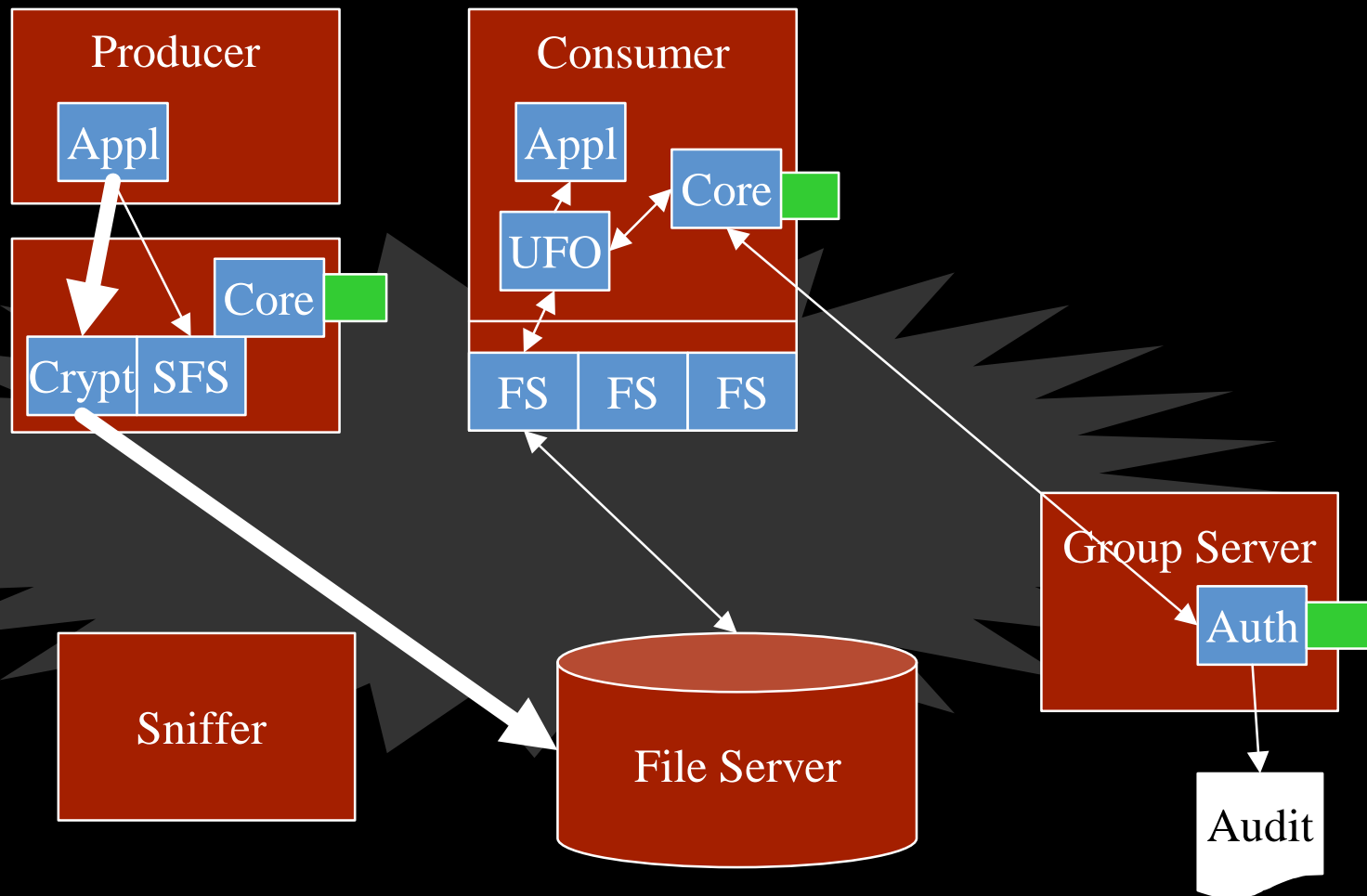
Production Configuration



Assured Configuration



High Performance Configuration



Information Security

- Protect Information
 - From Producer to Consumer
 - Not Infrastructure
- It is Information that has value
 - Not the networks
- Solve the insider threat
 - Thwart the hacker

<mailto:SFS@SecureFileSystem.org>
<http://SecureFileSystem.org>