

**Security 2.0
(An Approach to EKMI)**

Version 1.0

**Arshad Noor
CTO, StrongAuth, Inc.
arshad.noor@strongauth.com**

The 20th Century Internet

- University/DoD experiment
- From less than a dozen in the late '70s to perhaps some tens of millions in late '90s
- Mostly unprotected protocols & applications
- Shared-secret, Single-factor authentication
- Very little e-commerce

- Global phenomenon
- More than 0.6B devices, with > 0.1B being added each year
- Trust erodes with each new breach report
- Still using Shared-secret, Single-factor authentication
- Economy is increasingly dependent on the internet

- Hard shell, soft core
 - Focus on protecting the network
 - Detecting intrusions
 - Shared-secret, Single-factor authentication
 - Data is completely unprotected
 - Applications are security-ignorant
 - Little to zero user-education

- **Network security is NOT working**
 - PriceWaterhouseCoopers/CIO Magazine
Global State of Information Security Survey 2007
 - 7,200 CEOs, CFOs, CIOs, CSOs, VP's, Directors
 - 100 Countries
 - 36% North America
 - 28% Europe
 - 23% Asia
 - 12% South America
 - 2% Middle-East/South Africa

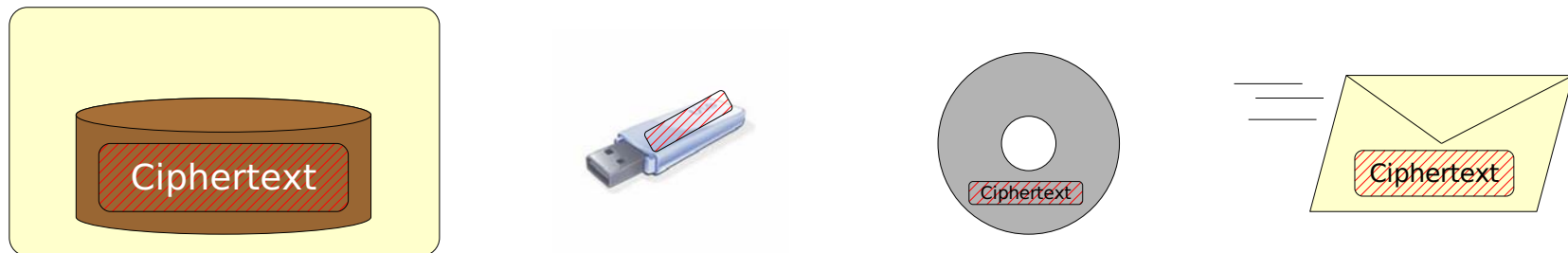
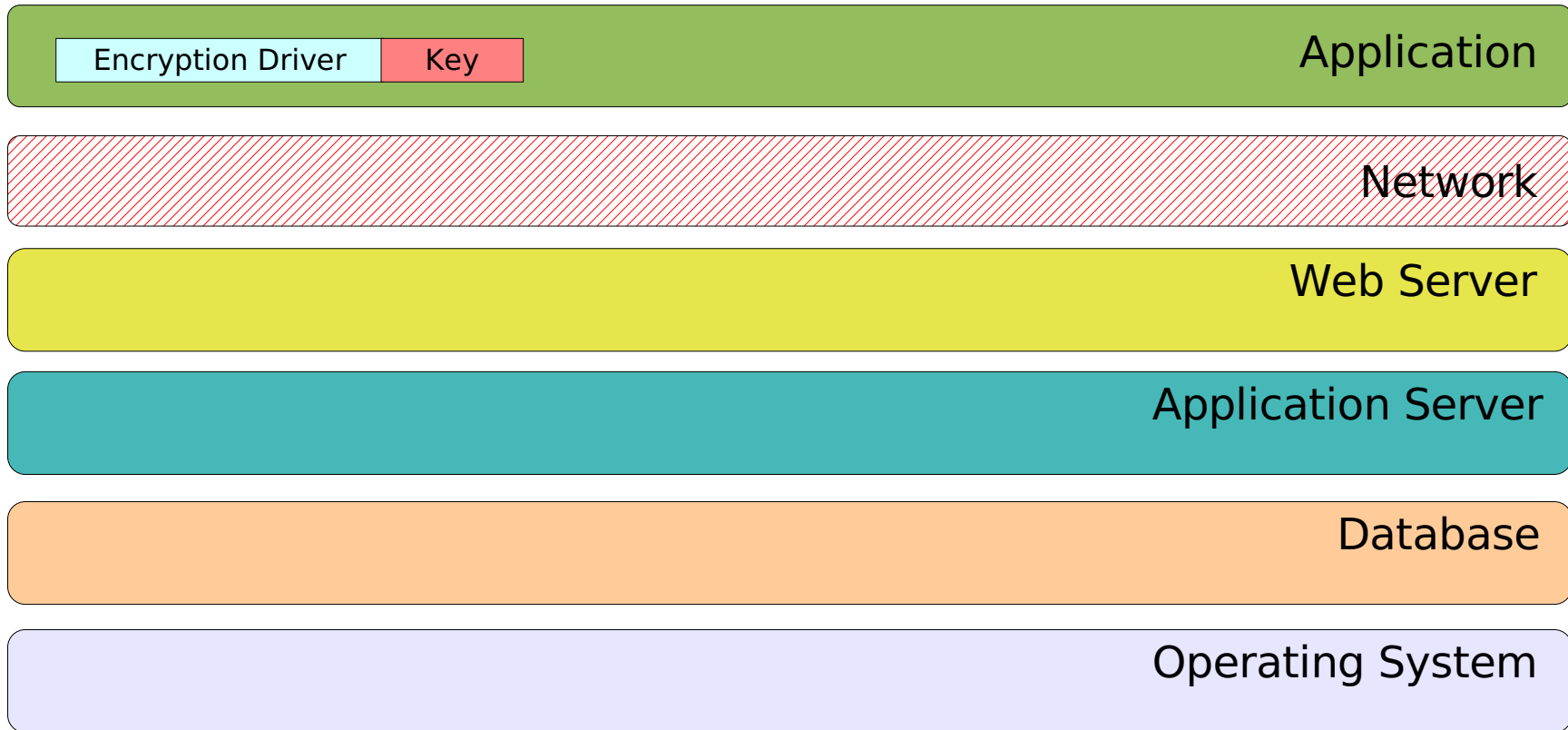
Source: <http://www.cio.com/article/133600/>

Security 1.0 Failures

- **40%** do NOT know how many security incidents they have experienced
- **45%** do NOT know what type of attacks have occurred
- **69%** do NOT keep an inventory of user data
- **67%** do NOT know where data is stored
- **33%** are NOT compliant with privacy laws

- Authentication should use IPF-6* as baseline
 - Asymmetric-key based authentication
 - External two-factor token
 - Sector-based credentials
- Application-independent authorization
- Data-protection should consist of encryption and integrity-protection *in the application*

* <http://middleware.internet2.edu/idtrust/2008/papers/01-noor-ipf.pdf>



The current problem



- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit



- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit



- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit



- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

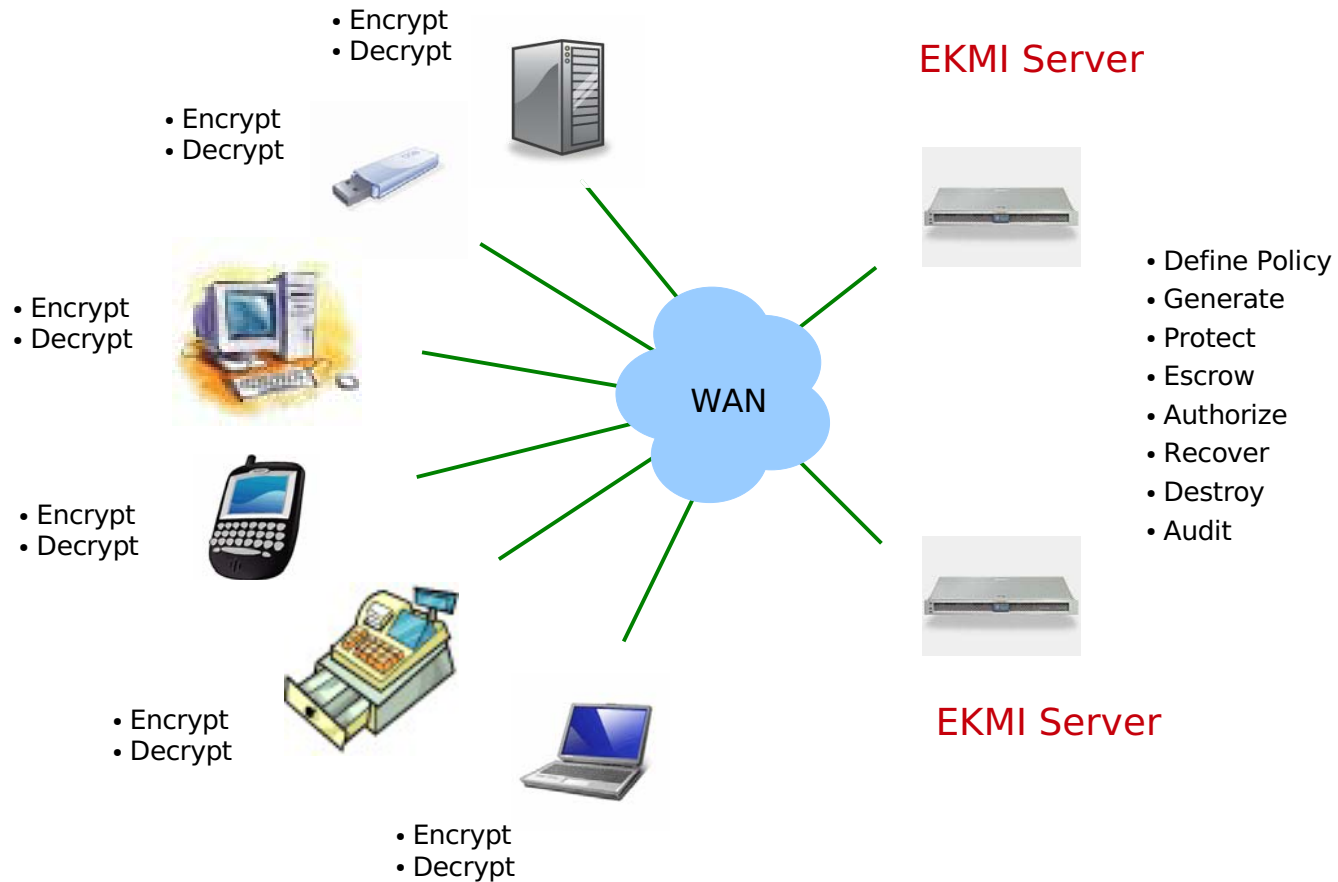


- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

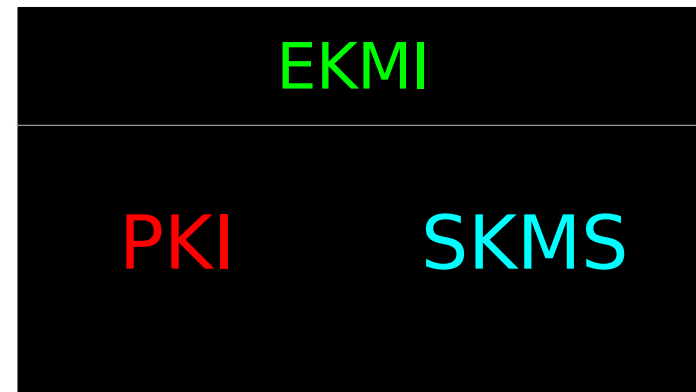


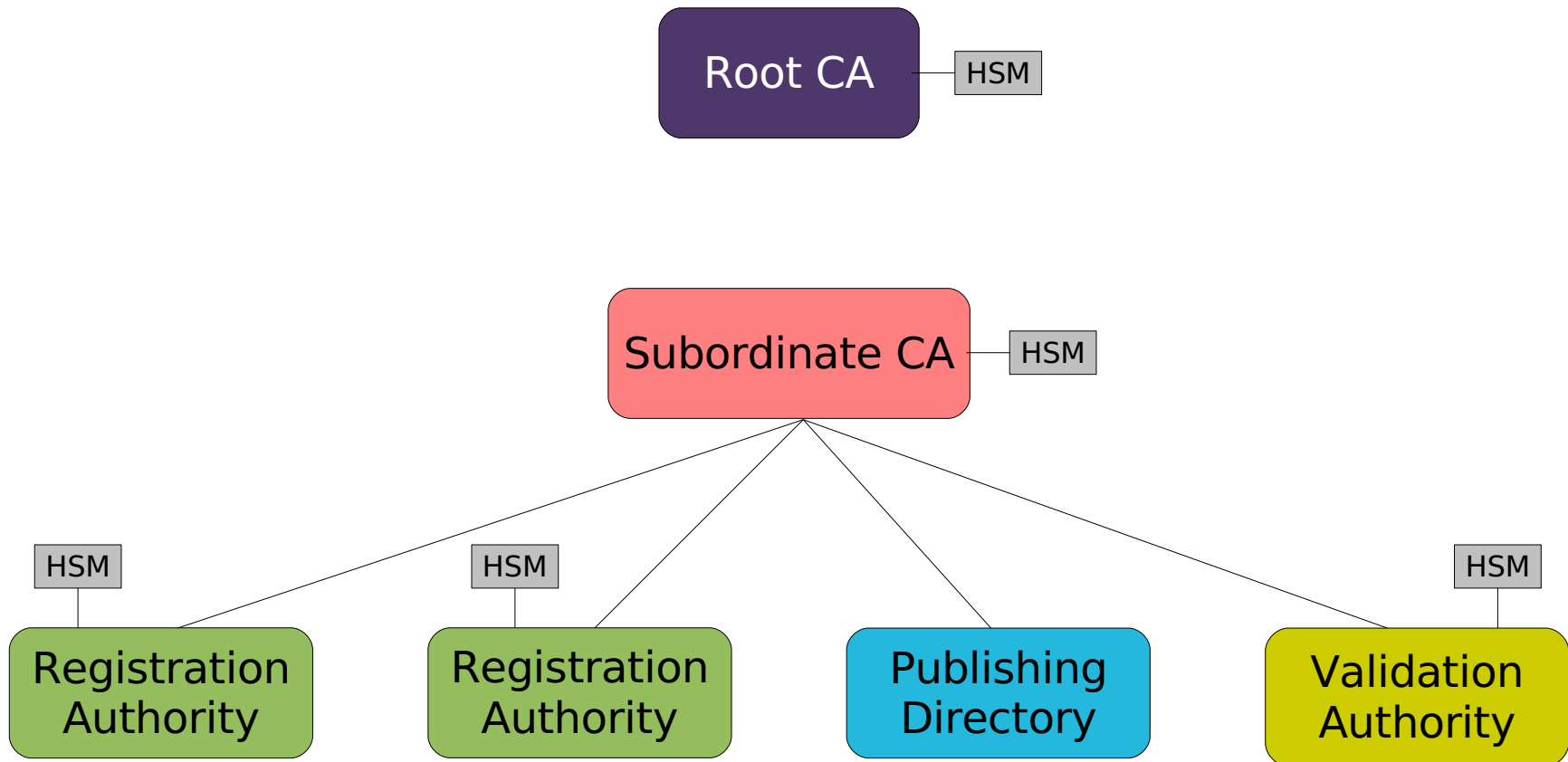
- Define Policy
- Generate
- Encrypt
- Decrypt
- Escrow
- Authorize
- Recover
- Destroy
- Audit

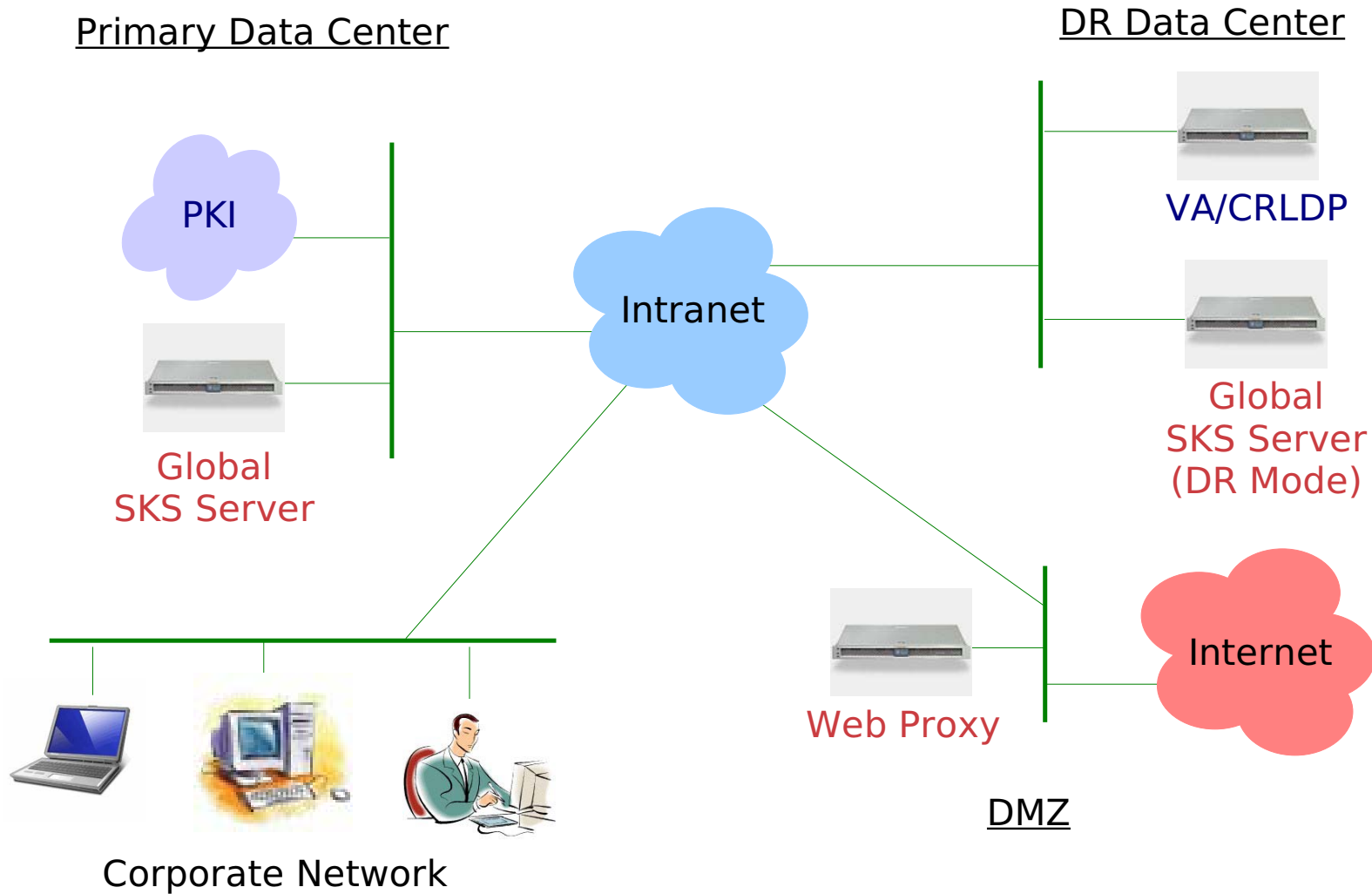
.....and on and on

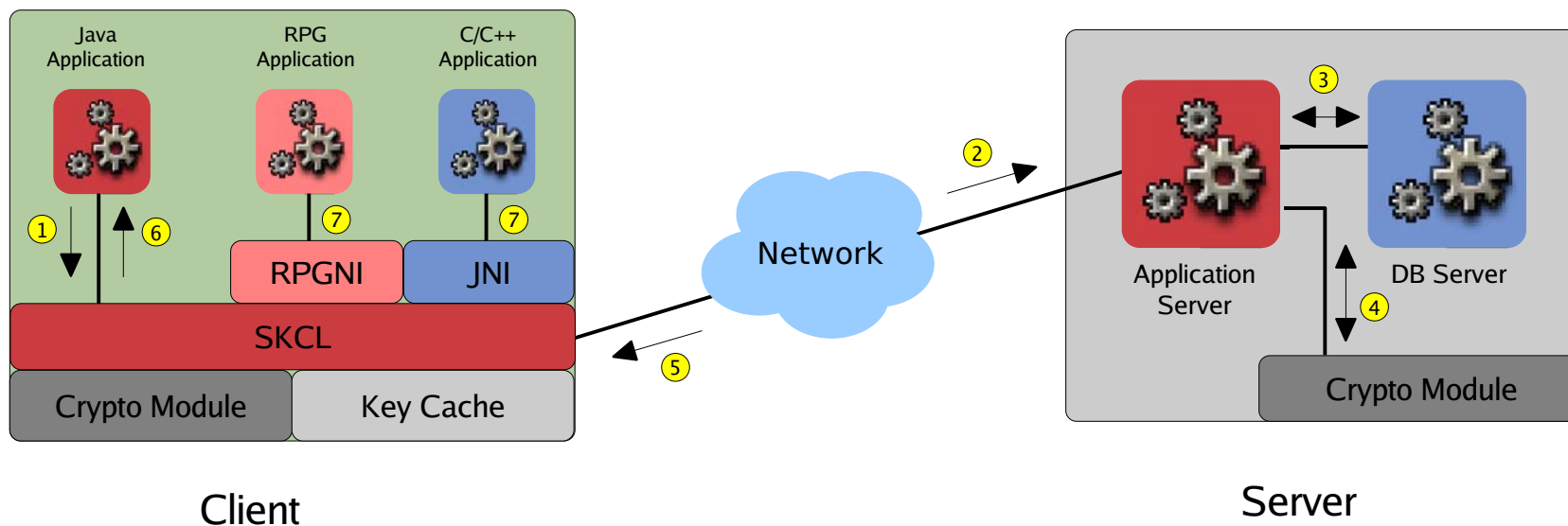


- Public Key Infrastructure (PKI)
- Symmetric Key Management System (SKMS)









1. Client Application makes a request for a symmetric key
2. SKCL makes a digitally signed request to the SKS
3. SKS verifies SKCL request, generates, encrypts, digitally signs & escrows key in DB
4. Crypto HSM provides security for Signing & Encryption keys of SKS
5. SKS responds to SKCL with signed and encrypted symmetric key
6. SKCL verifies response, decrypts key and hands it to the Client Application
7. Native (non-Java) applications make requests through Java Native Interface

- Low-cost
 - Open-Source PKI and SKMS
- Fast
 - Choice of OS, DB, HSM, etc.
- Secure
 - Security 2.0 features in the EKMI itself
- Management
 - Key-caching, Replication, Centralized, etc.

- Questions?
- Contact Information
 - www.strongauth.com
 - www.strongkey.org
 - info@strongauth.com
 - (408) 331-2000