Introduction to the GST Security Architecture

Danny De Cock

SEVECOM Workshop

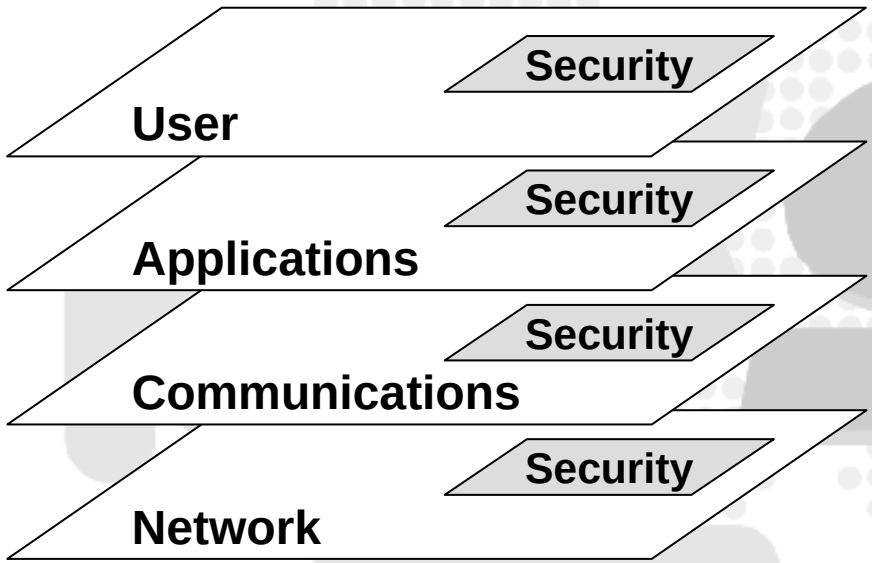
4-5 September 2006

Budapest, Hungary

SEC Goal

- **Define an architecture and provide security mechanisms for secure telematics applications**
  - **Functional point of view**
    - **Applications, services, user devices…**
  - **Infrastructure point of view (networks, platforms)**

Security – Where?



Focus

Strong authentication of {user, device, service provider} ✓

Applications integrity

Secure communications ✓

Network access

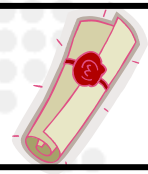
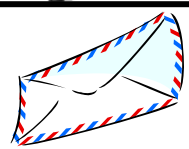
User Requirements:

- **"I do not want to pay for services I did not order/use"**
  - **Authentication and Non-repudiation**
- **"I do not want that unauthorized parties are able to monitor what I do"**
  - **Privacy and confidentiality**

Security – How?

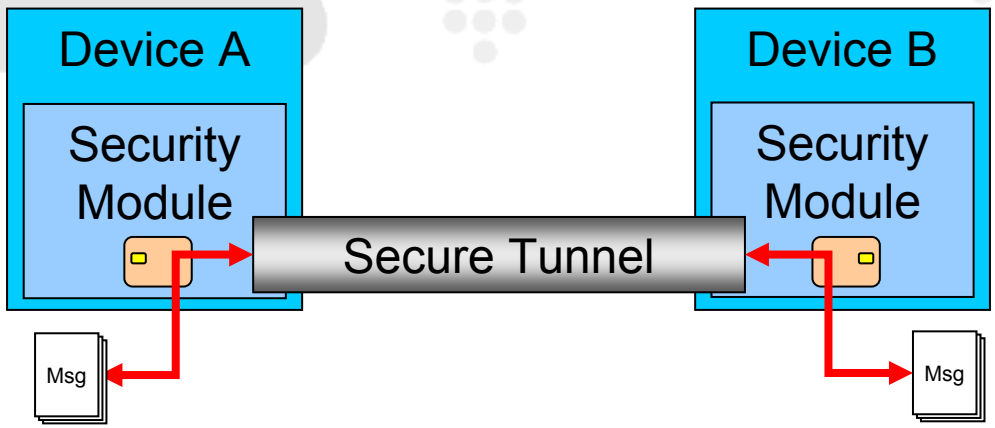
Based on implementation complexity and cost:

- No security mechanisms
- Non-cryptographic techniques (e.g., CRC, hardware enclosures,…)
- Combine all of the above with cryptographic techniques

Security Levels		Protect Confidentiality	
		Yes	No
Protect Integrity	Yes	Secure 	Authenticated 
	No	Confidential 	Insecure 

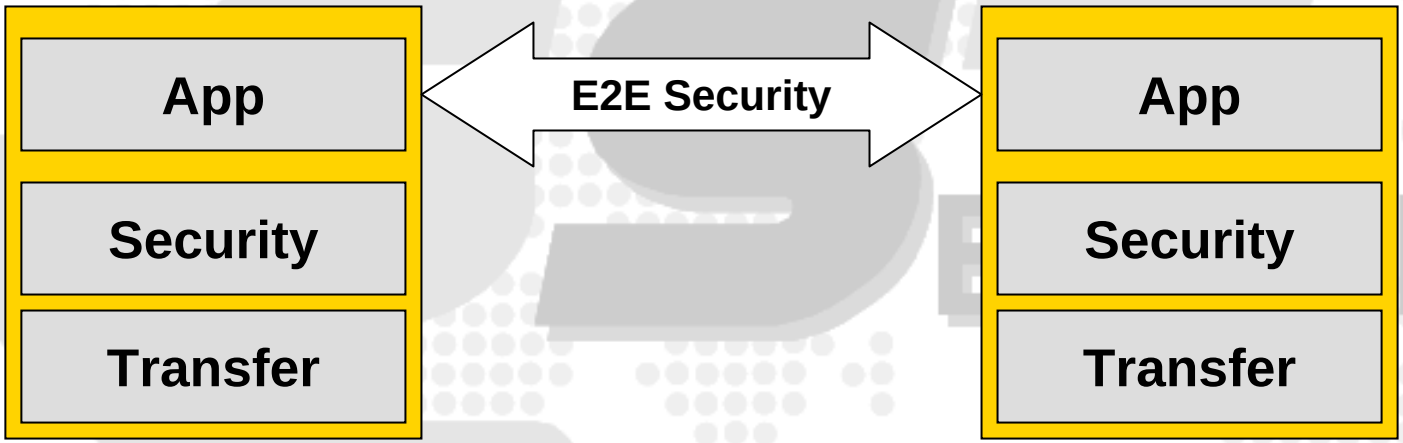
Security – What?

- User/Services data
  - User requests service
    - Information and data exchange
  - Service provider provides service
    - Client-server model
- Application data
  - Sent between service provider and device

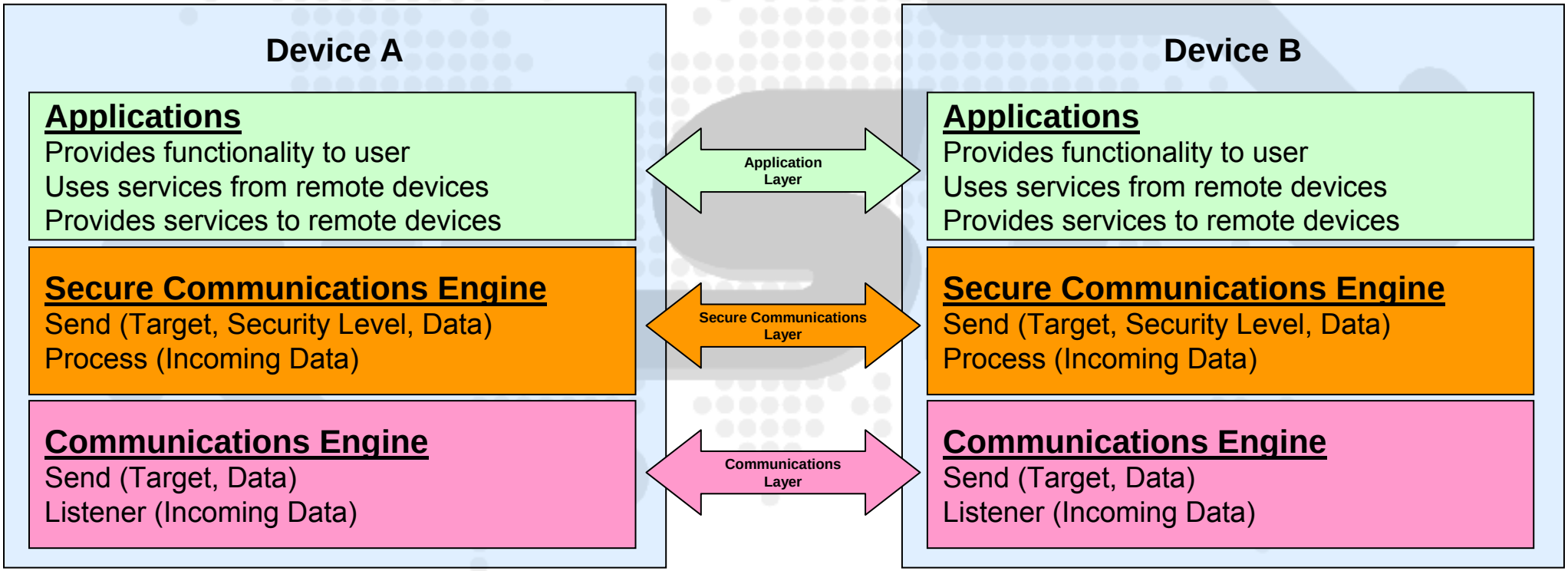


SEC Impact

- Reference points follow a layered model

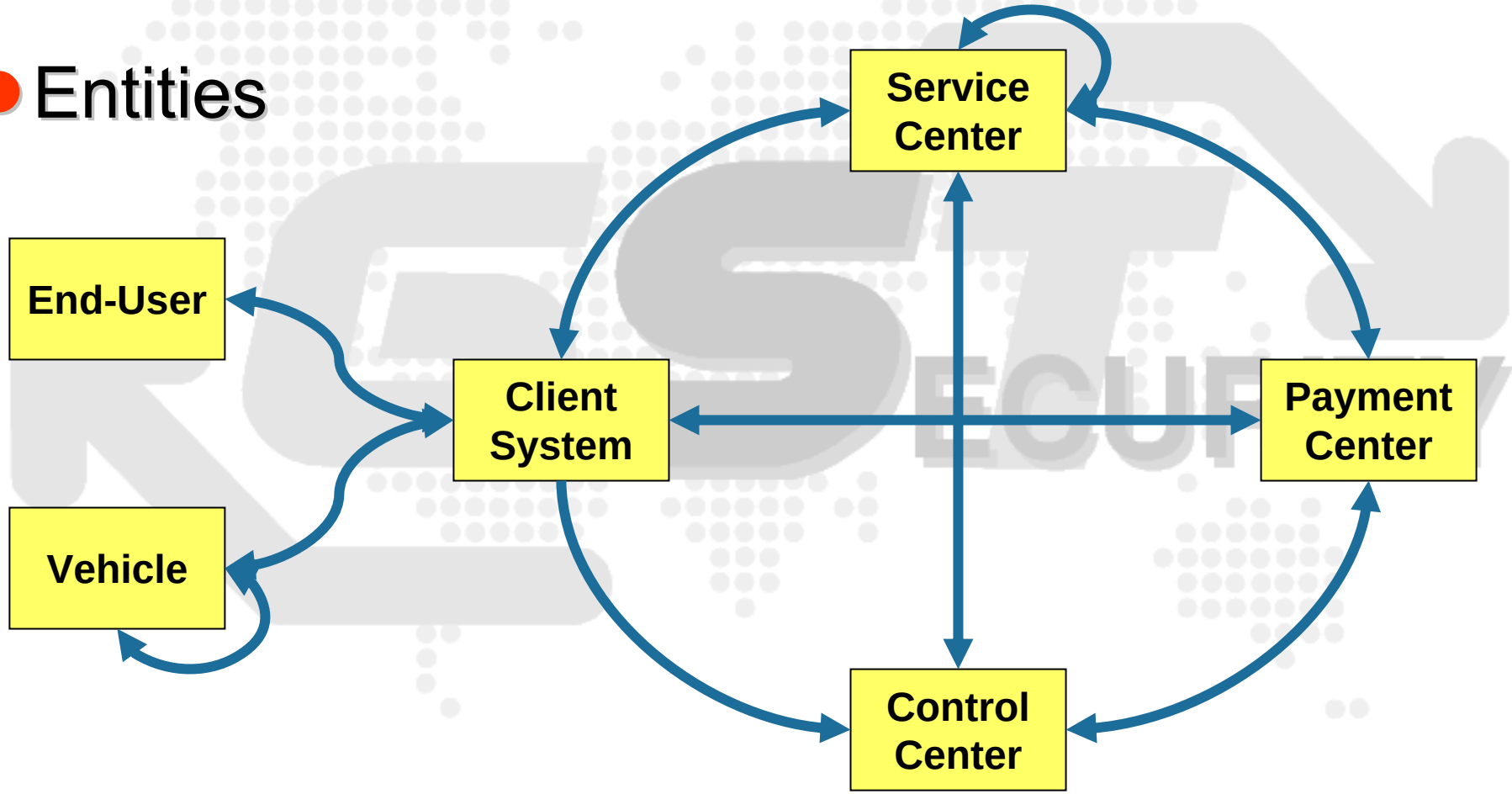


{Device, Infrastructure} → {Infrastructure, Device}

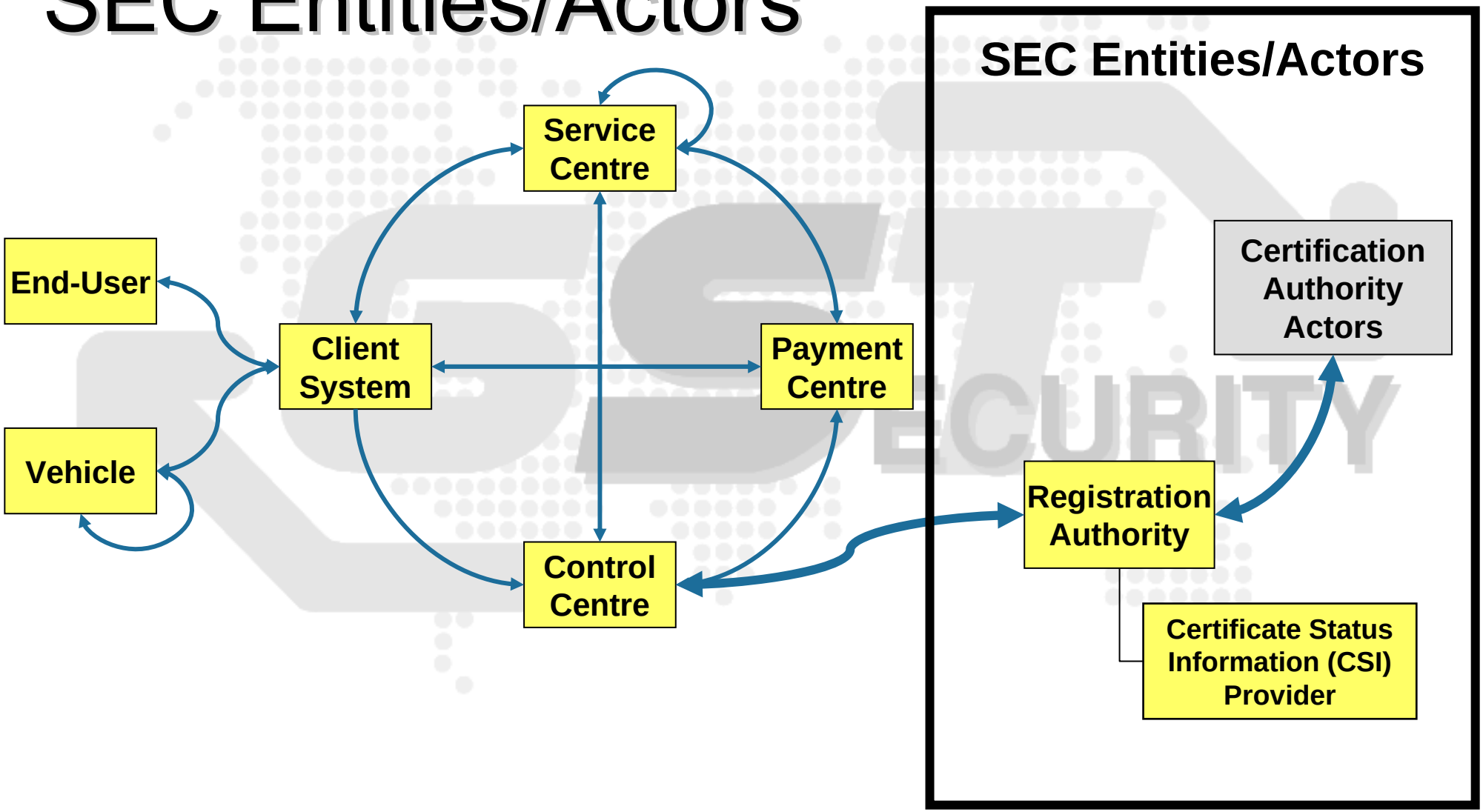


GST High-level Architecture

- Entities

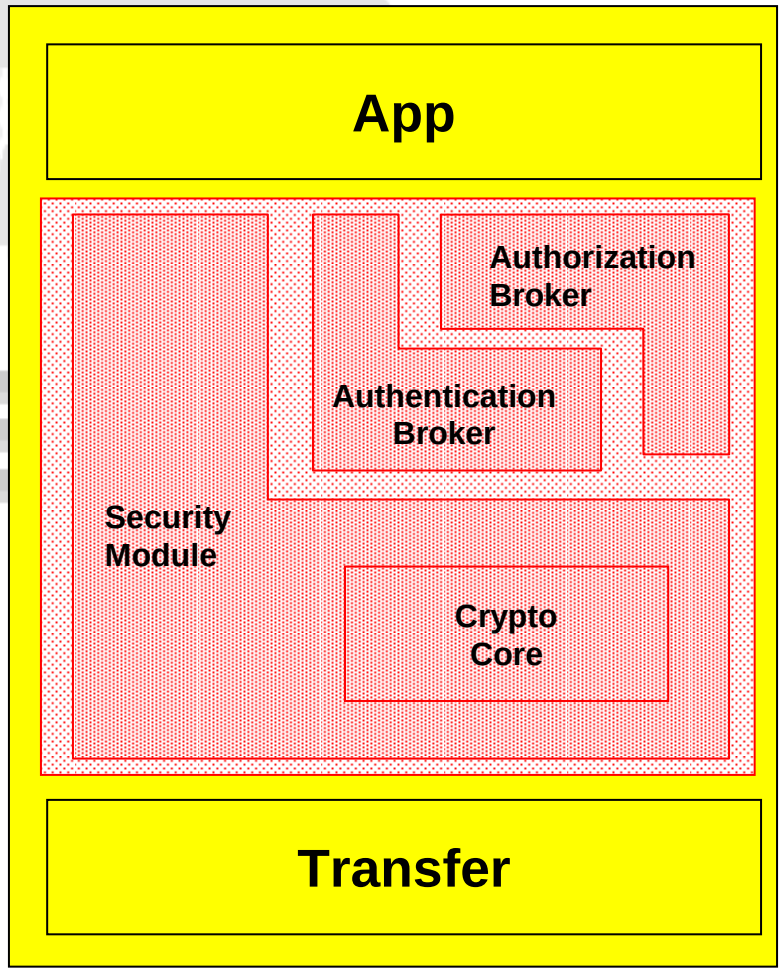


SEC Entities/Actors

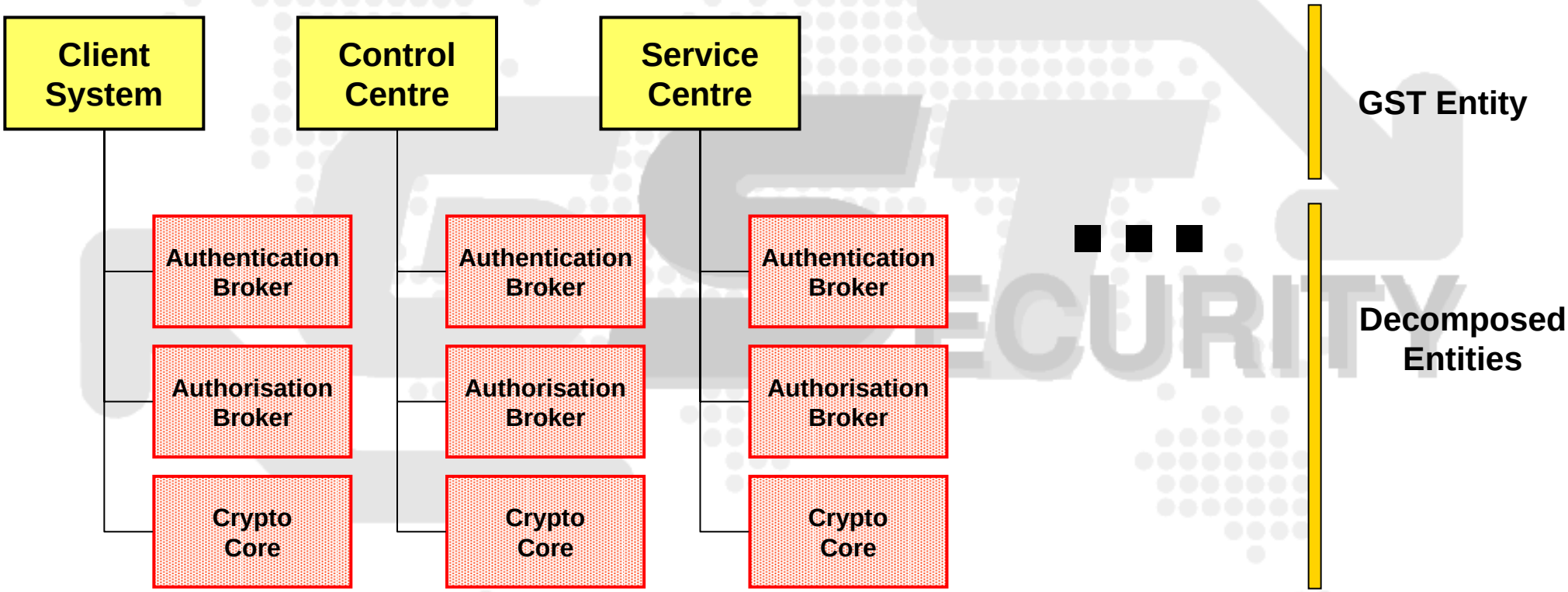


Security Aware GST Entity

- Has an authorization broker
  - Validates whether certain actions can be allowed, e.g., incoming network traffic, software update,…
- Has an authentication broker
  - Validates authenticity of other GST entities and End users
  - Authenticates data sent from this entity to another GST player
- Includes a security module
  - Stores the entity's credentials (e.g., session keys, trusted certificates,…)
  - Protects confidentiality and/or integrity of persistently stored data (e.g., log files, user data, system data,…)

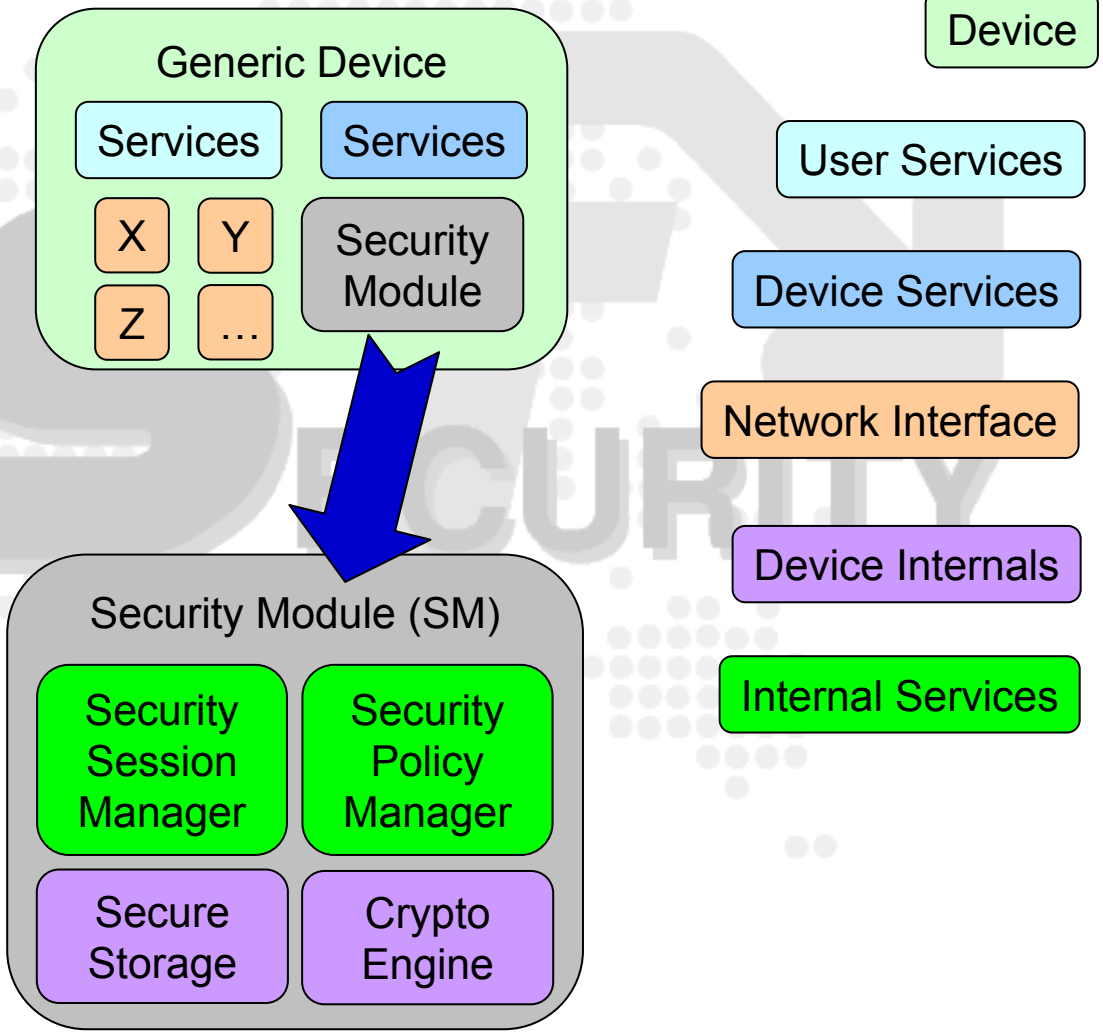


SEC Decomposed Entities



Devices and Security Modules

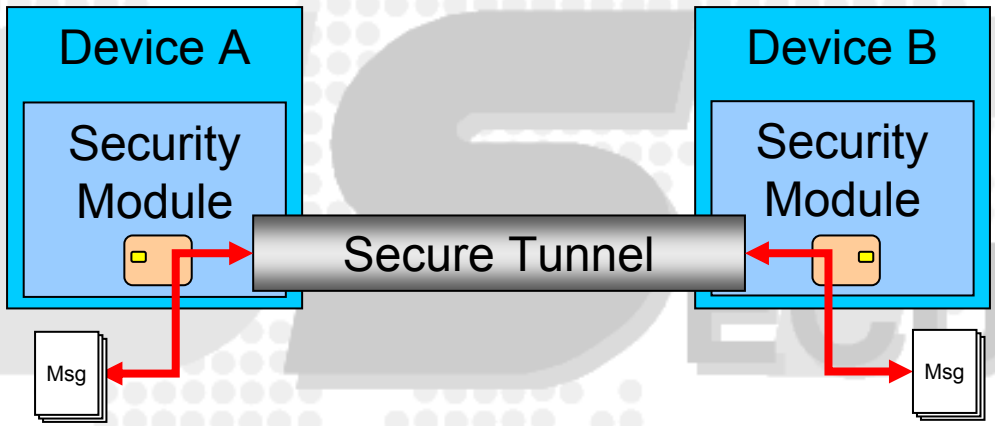
Key Features of a Security Module:
- One SM per Device
- SM = e.g., OSGi bundle
- SM offers services to other bundles
- SM initialized by manufacturer
- Initialized SM ready to be used
- Combination of hard- and software
  - Hardware ➔ Non-cloneable
  - Software ➔ Risk for cloning
- Provide true strong authentication
- Secure communications rely on SM
  - Insecure
  - Authenticity
  - Confidentiality
  - Secure = Auth. + Conf.
  - Optionally: Anonymous

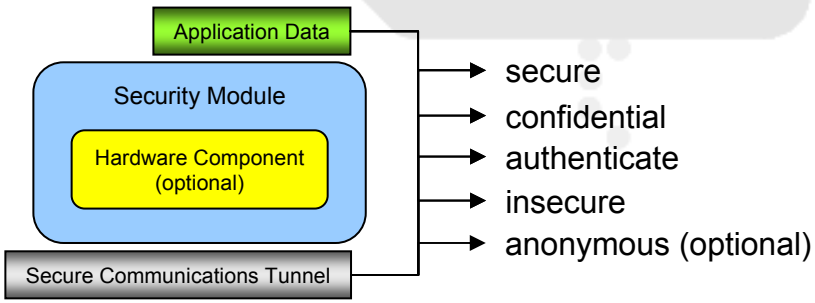


SM Functionality

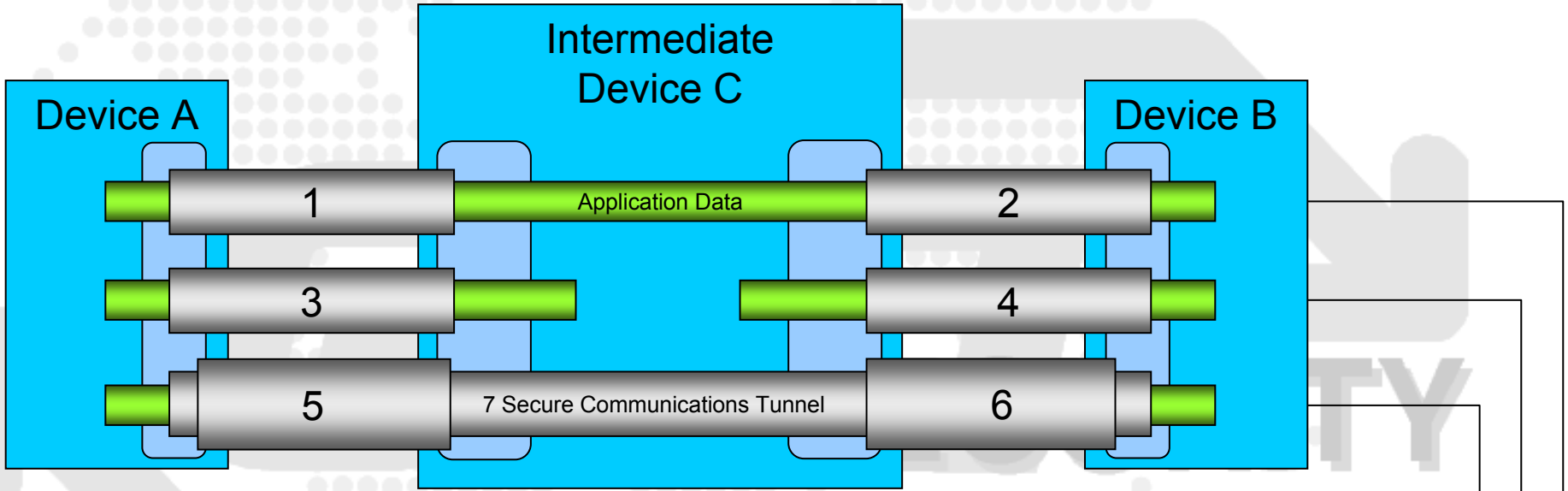
- Access-controlled/Secure persistent storage engine
  - User data, Communications session data
- Authentication engine
  - Digitally sign outgoing information
  - Calculate Message Authentication Code (MACs)
  - Verify incoming authenticated data
- System-wide “trusted” information
  - Root CA certificates
  - Trust anchors with respect to registration proofs
- Operates in client-server mode
  - Difficult to enforce use of security module at client side
  - Server can determine whether the correct SM was used

Secure Communications





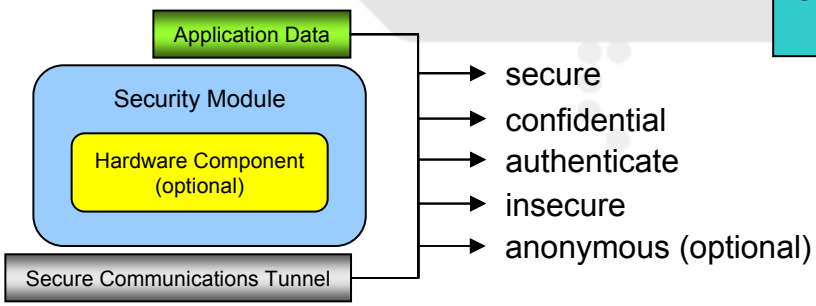
Secure Communication Types



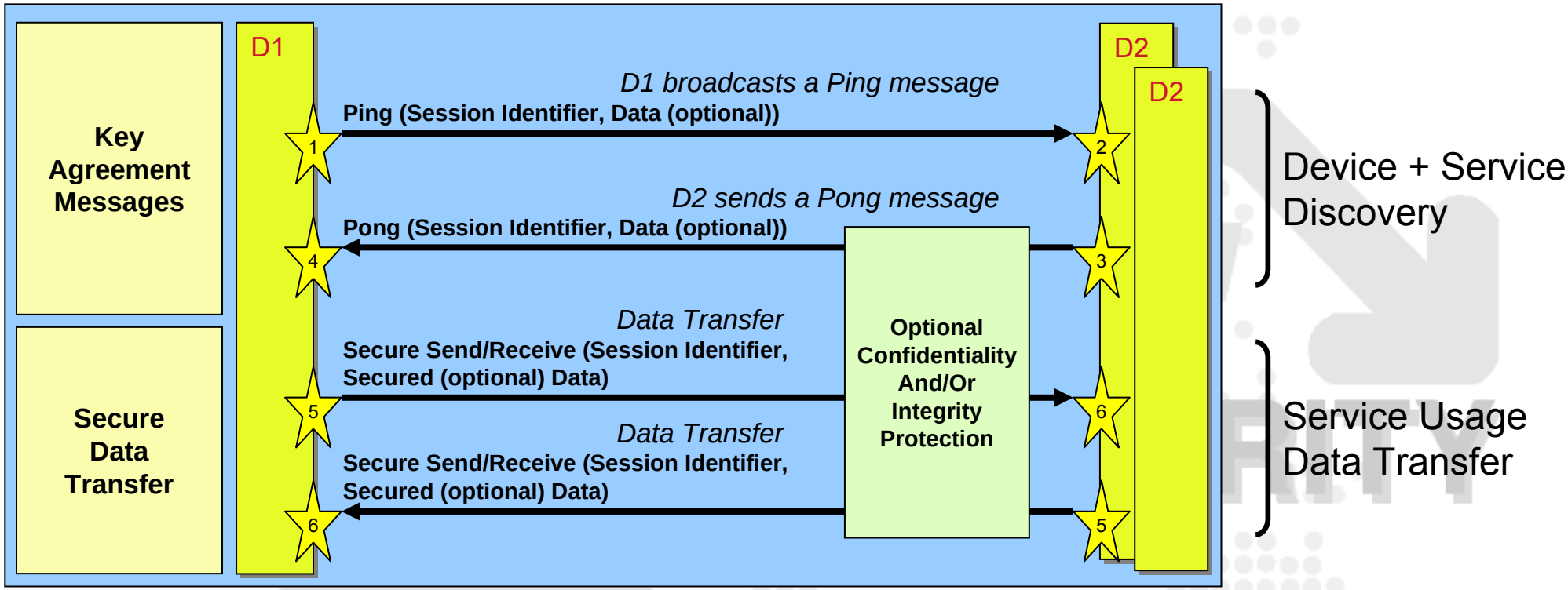
Store and Forward of Application Data through End-to-End secure tunnel
Full security options, heterogeneous security mechanisms

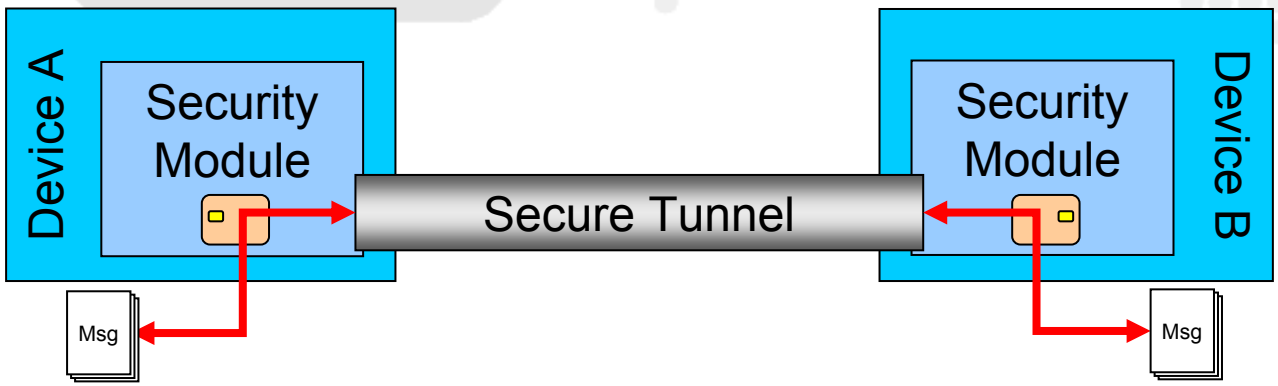
Application Data, Point-to-Point Secure Communications

Application Data Securely Exchanged Between A and B



Secure Key Agreement with Station-to-Station





Thank you for your attention

Danny De Cock

Danny.DeCock@esat.kuleuven.be

http://www.esat.kuleuven.be/cosic

GST – Global System for Telematics

http://www.gstforum.org

"If it is provably secure, it is probably not…" – Lars R. Knudsen on block ciphers

Secure Key Agreement with Station-to-Station (ctd)

1

Ping message sent from D1 to D2
- Computes secret *x*
- Calculates α^x
- Authenticates $\{data_1||\alpha^x\}$

D1 Broadcasts the Ping message
- Broadcast of Authenticated $(data_1||\alpha^x)$

2

D2 Receives a Ping message
- Checks Authenticated $(data_1||\alpha^x)$
- Processes $data_1$

3

D2 Prepares a Pong message for D1
- Computes secret y
- Calculates α^y
- Calculates $K= (\alpha^x)^y$
- Encrypts data: $E_K(data_2)$
- Authenticates $\{E_K(data_2)||\alpha^y\}$

D2 Broadcasts Pong message for D1
- Broadcast of Authenticated $(E_K(data_2)||\alpha^y)$

4

D1 Receives a Pong message
- Checks Authenticated $(E_K(data_2)||\alpha^y)$
- Calculates $K= (\alpha^y)^x$
- Decrypts $E_K(data_2)$
- Processes $data_2$

5

D1 Prepares Secure Data Transfer
- Encrypts $E_K(data_3)$
- Authenticates $E_K(data_3)$

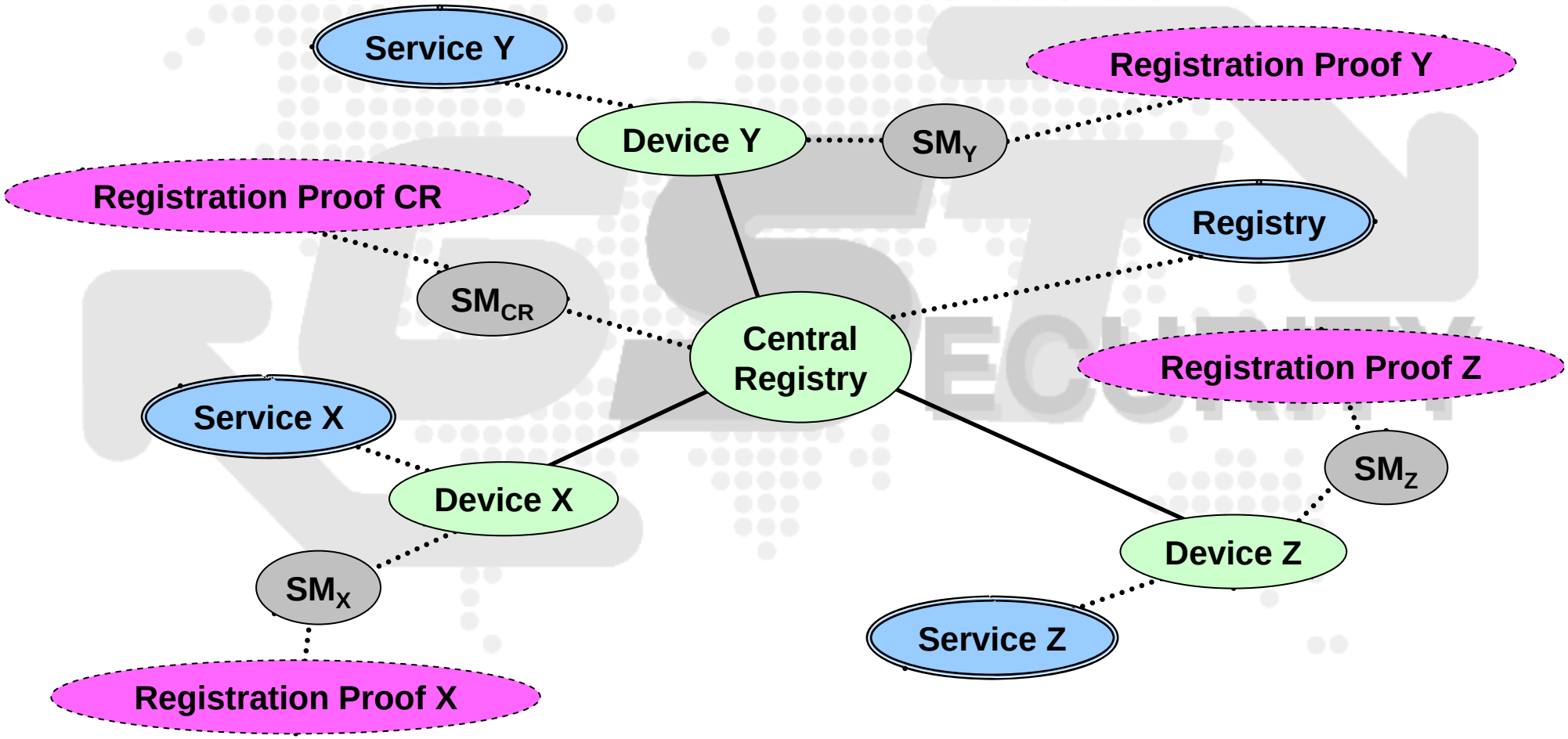
D1 Broadcasts Secured Data Transfer message for D2
- Broadcast of Authenticated $(E_K(data_3))$

6

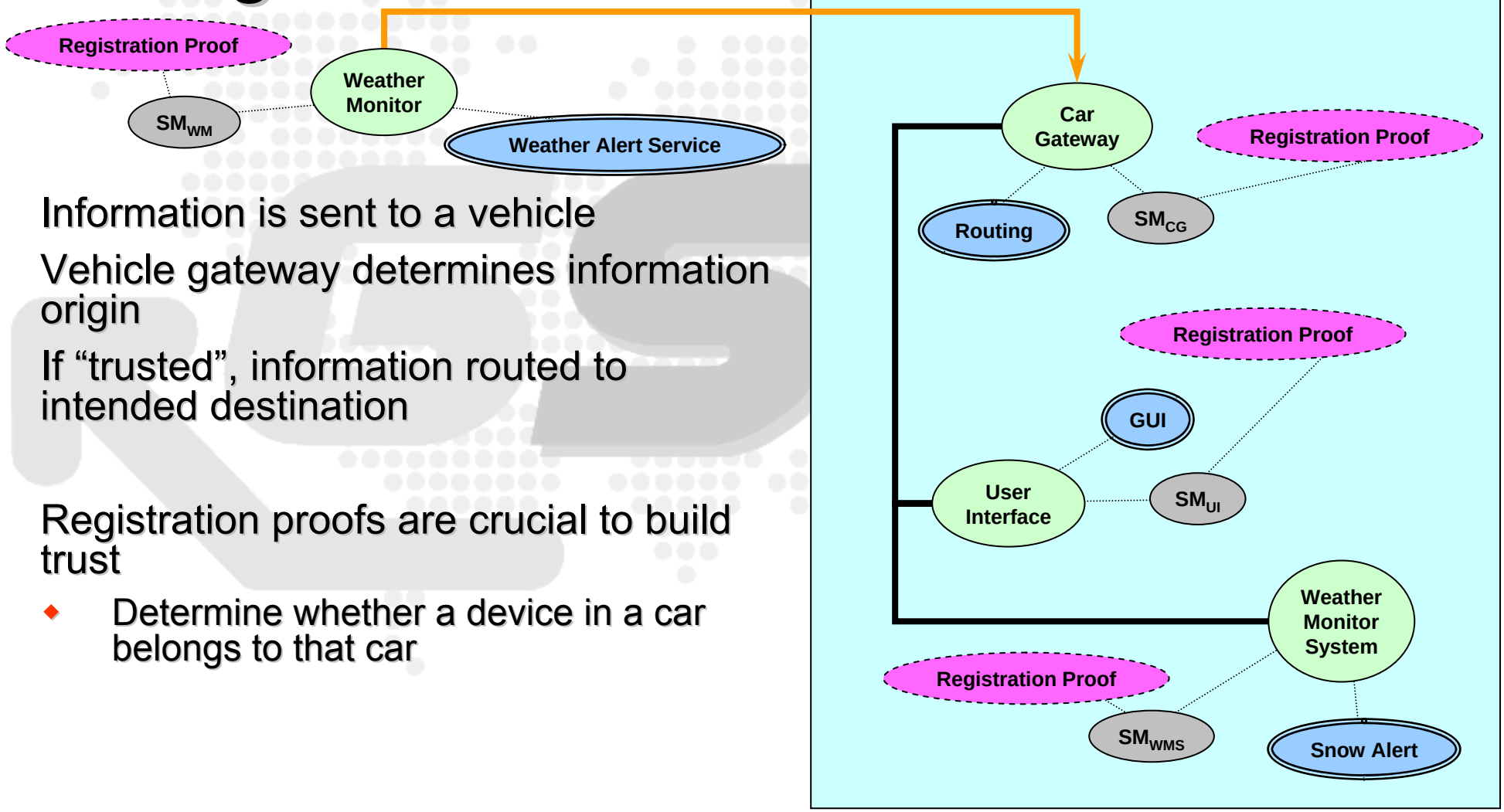
D2 Receives a Secured Data Transfer message
- Checks Authenticated $(E_K(data_3))$

D2 Decrypts the information within a session with D1
- Decrypts $E_K(data_3)$

Devices Registration



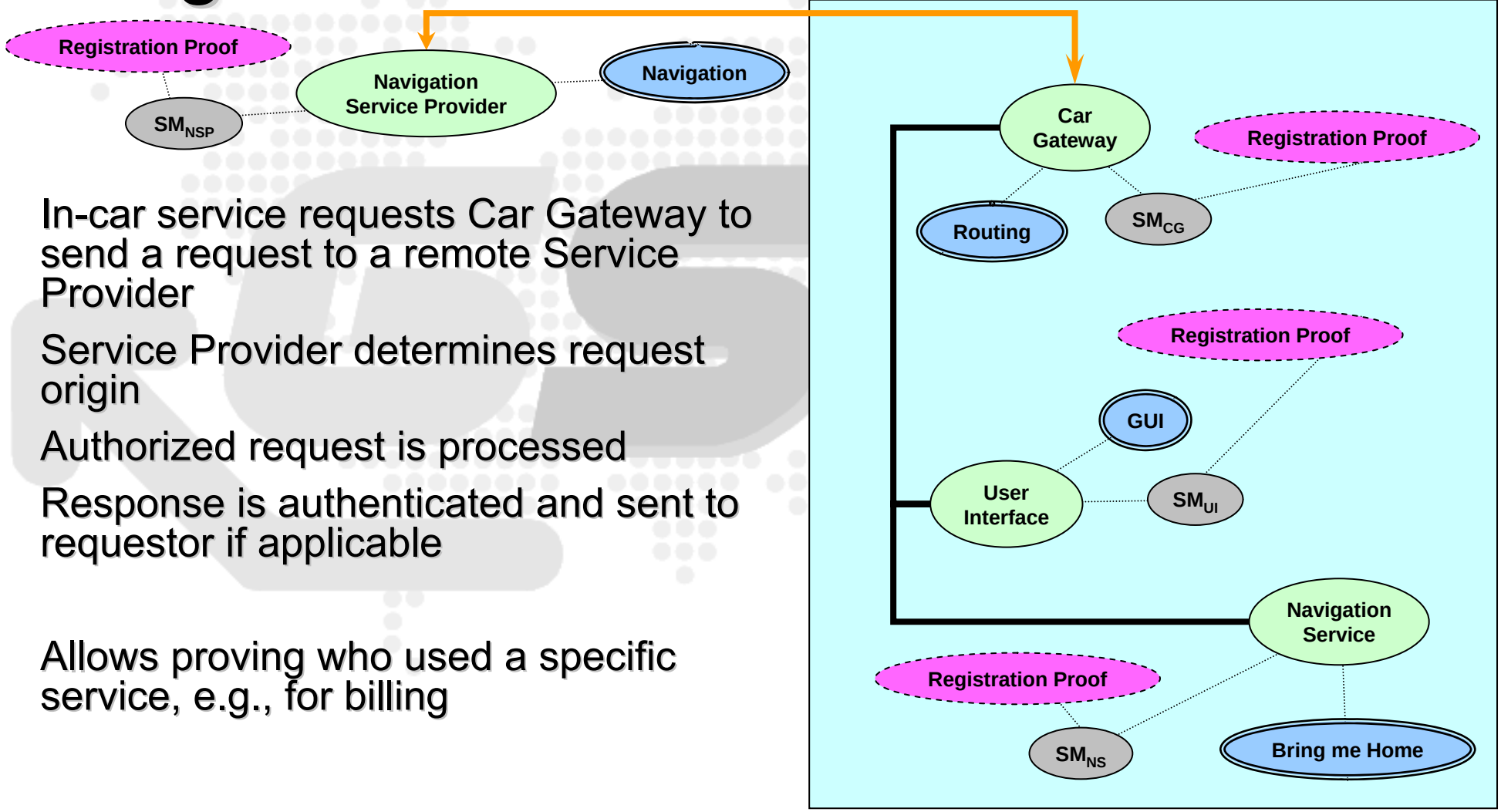
Pushing Data to Car



- Information is sent to a vehicle
- Vehicle gateway determines information origin
- If “trusted”, information routed to intended destination

- Registration proofs are crucial to build trust
  - Determine whether a device in a car belongs to that car

Pulling Data to Car



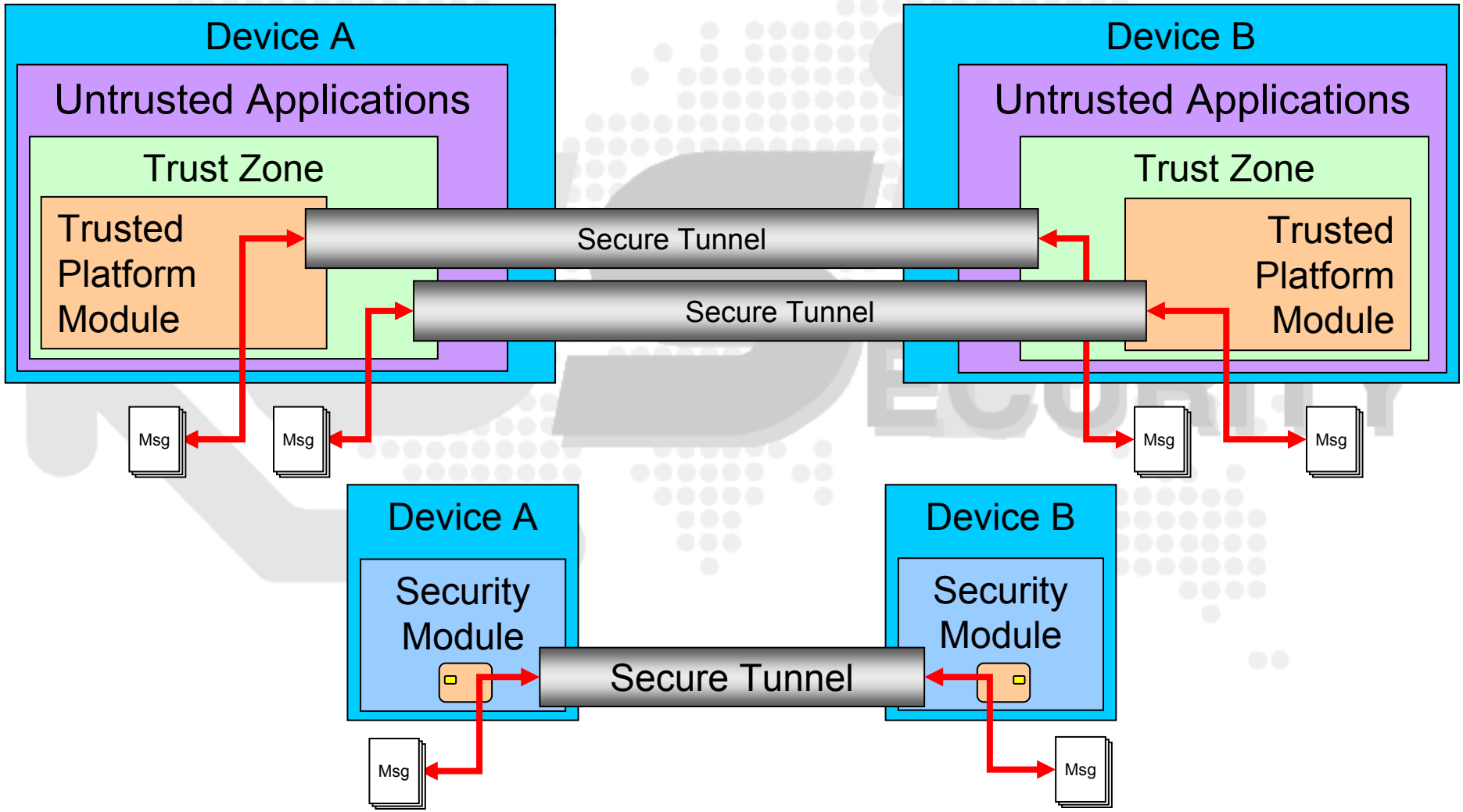
- In-car service requests Car Gateway to send a request to a remote Service Provider
- Service Provider determines request origin
- Authorized request is processed
- Response is authenticated and sent to requestor if applicable

- Allows proving who used a specific service, e.g., for billing

Examples of Security Modules

- Hardware security module (most expensive)
  - Used for high-bandwidth communications, secure payments, etc.
  - Not very car-friendly ☺
- Smartcard, SecurID token, SIM card
  - Commonly used to provide strong authentication
  - Reasonably cheap
- Trusted platform module (TPM)
  - By default built into many new laptops and desktops
  - Cheap
- Software key store (cheapest)
  - Less critical applications

Relation with Trusted Computing



Protocol Stacks View

User Layer
User devices

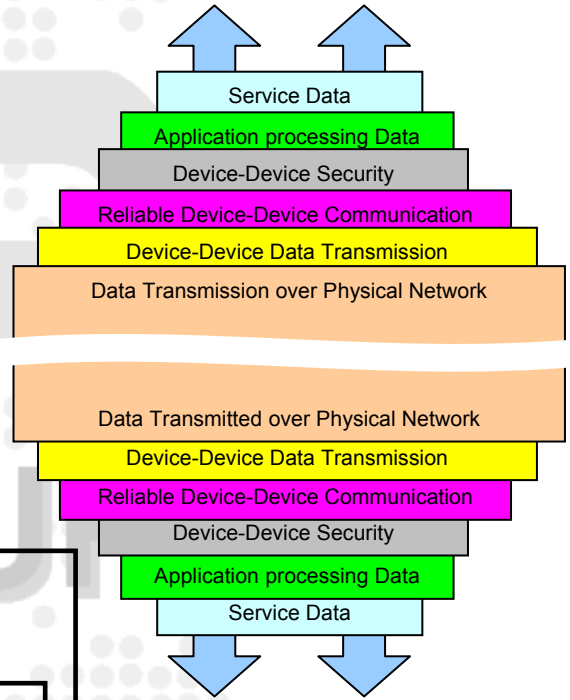
Application Layer (OSI Layer 7)
Offers Services to Users, Services and Devices

Security Layer (OSI Layer 5 – Session)
Protects Against Remote Evil Services and Devices

Transport Layer (OSI Layer 4)
Provides Reliable Communications (order, arrival,…)

Network Layer (OSI Layer 3)
Provides Network Access, e.g., ETH, ZigBee,…

Data Link Layer (OSI Layer 2)
Communication Technologies, e.g., RF, WiFi, IR,…



Our Security Features

SSL/TLS, VPN…

SSH, IMAP, IMAPS, HTTP…

ISO 15764, ISO/DIS 20828

IPSEC

Typical Examples