A Resilient Protection Device for SIEM Systems

Nuno Neves

University of Lisboa, Portugal

http://www.di.fc.ul.pt/~nuno

Work with: Alysson Bessani, Miguel Garcia, Eric Vial, Ricardo Fonseca, Paulo Veríssimo

EC project FP7-257475

MASSIF: MAnagement of Security information and events in Service InFrastructures

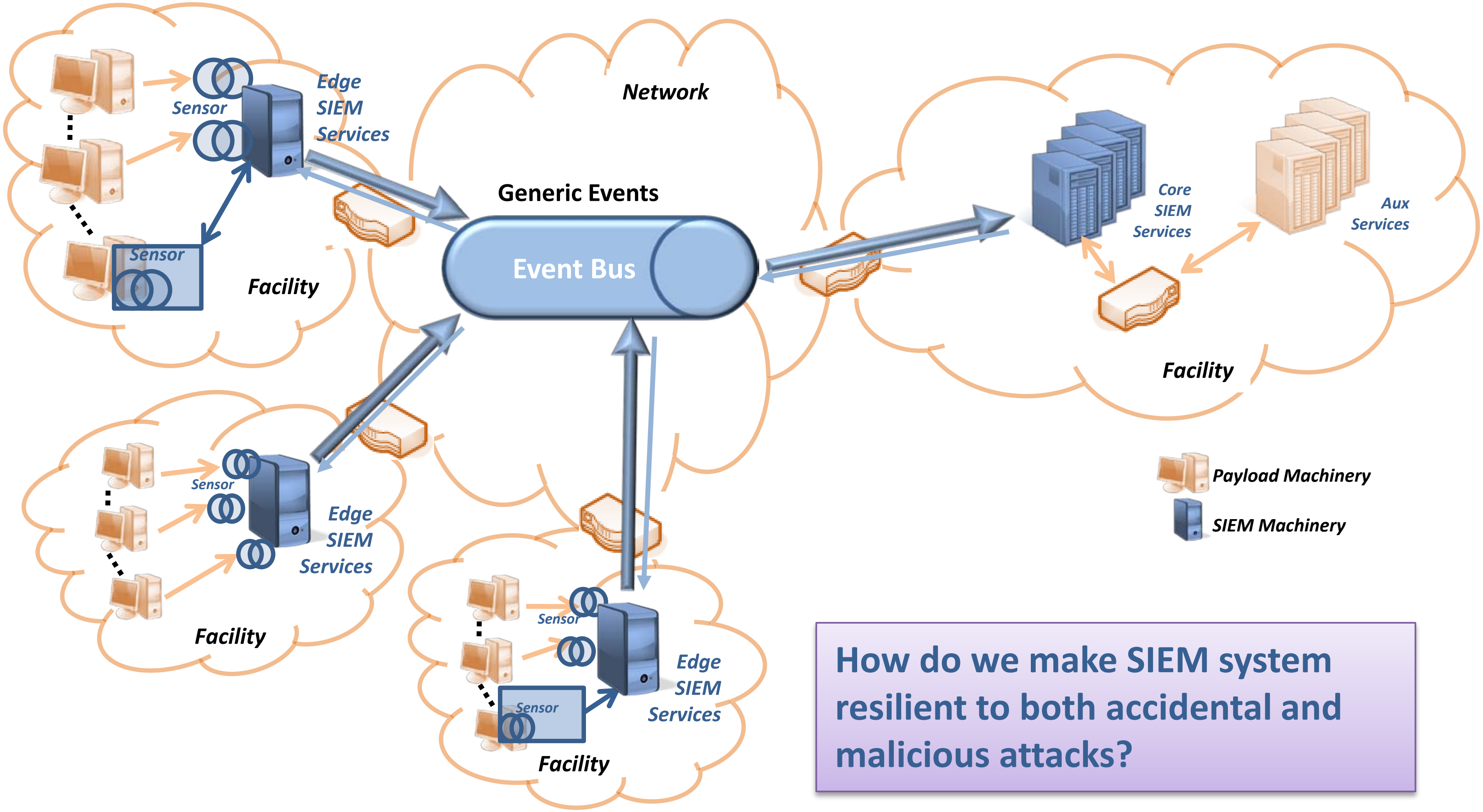
Project funded by the European Commission ICT workprogramme 2009 (FP7-ICT-2009-5)

SEVENTH FRAMEWORK PROGRAMME

Context

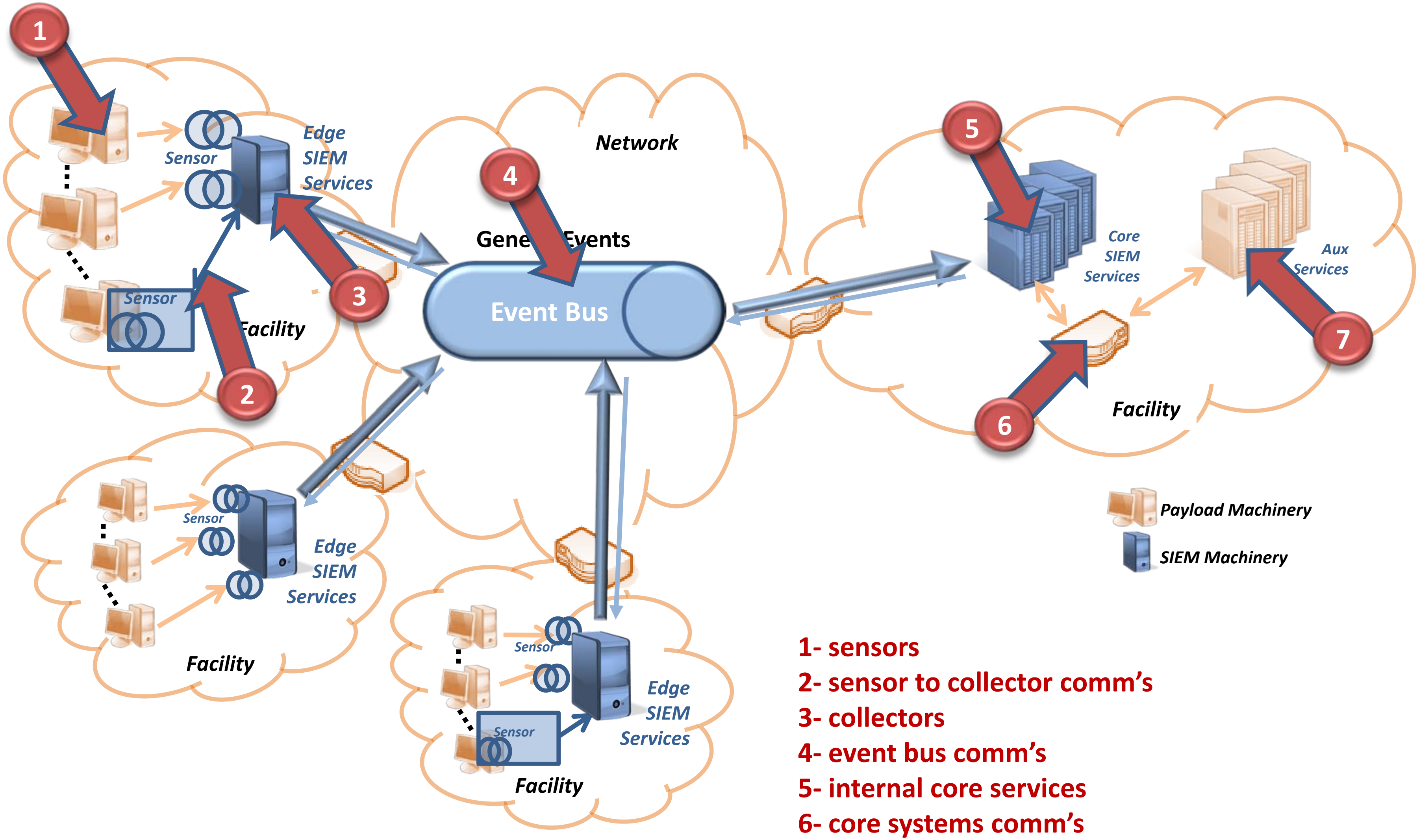
- Security Information and Event Management (SIEM) systems offer various capabilities for the
  - collection and analysis of security information in networks
  - allowing the correlation of thousands of events and the reporting of attacks and intrusions in near real-time
- Main components
  - Sensors: collect information about the local environment and help on the responses; Can be: signature or anomaly-based IDS; vulnerability scanners; network profiling; inventory management
  - Collectors: gather and normalize the events generated by the sensors and any external systems; can be deployed standalone or in a Sensor
  - Management server (or SIEM engine): event correlation and real-time monitoring; risk assessment; reporting and data mining; network profiling and inventory management

SIEM Architecture – A Structural View



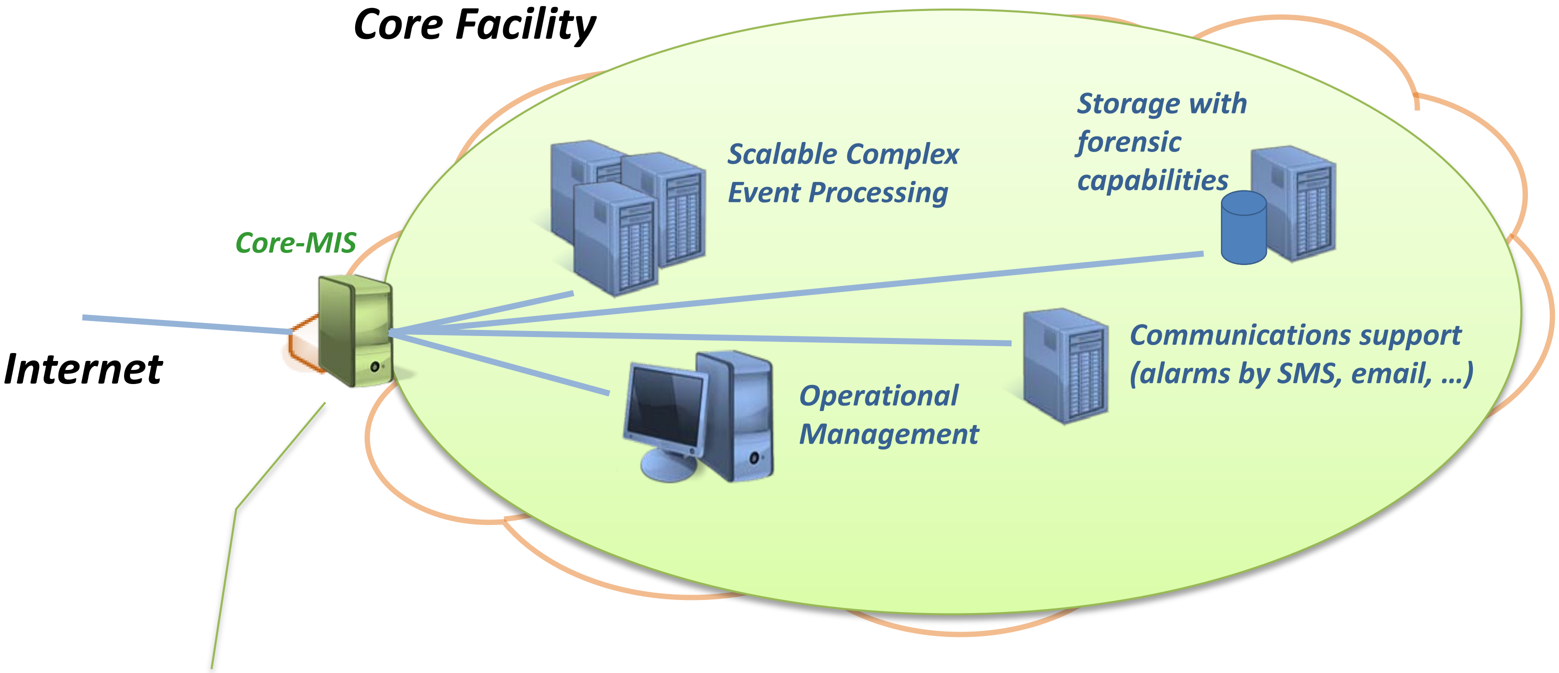
How do we make SIEM system resilient to both accidental and malicious attacks?

SIEM Architecture – Attack Vectors



1- sensors
2- sensor to collector comm's
3- collectors
4- event bus comm's
5- internal core services
6- core systems comm's
7- auxiliary core system

Protecting the Core Services


Core Facility
Storage with forensic capabilities
Scalable Complex Event Processing
Core-MIS
Internet
Communications support (alarms by SMS, email, …)
Operational Management

Like a highly resilient application-level firewall protecting the information flows in and out of the core SIEM services

Application Firewalls / IPS

- Acts on the application layer, inspecting the traffic to block malicious content that attempts to exploit known logical flaws at the destination
- May also offload encryption from servers, block application input/output from detected intrusions or malformed communication, manage or consolidate authentication

Security related vulnerabilities, OSVDB, 2010 to 2012

Barracuda Networks	29
Cisco Adaptive Security Appliance	36
Juniper Networks	64

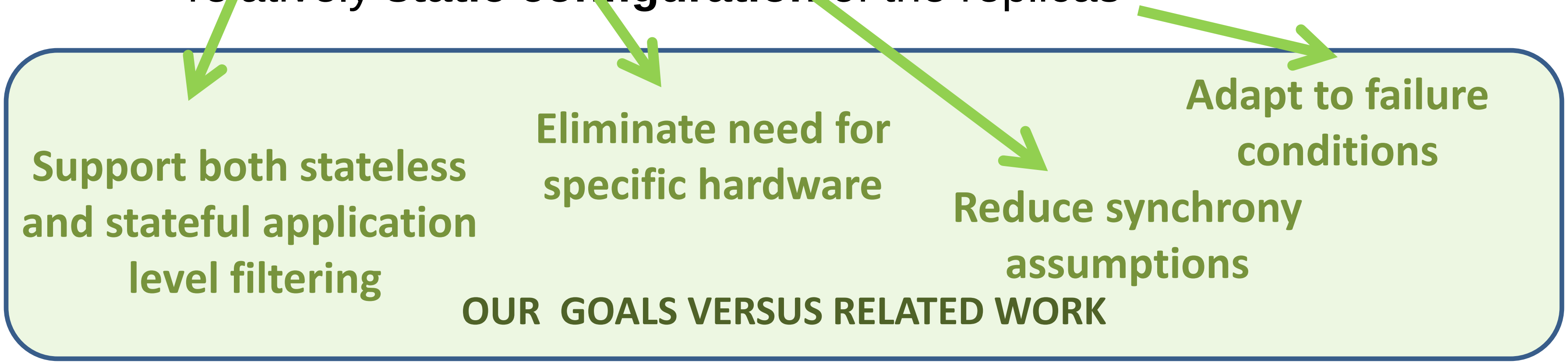
Packet filter	3
Comodo Internet Security	10
netfilter	29

Cisco Adaptive Security Appliances contains a flaw that may allow a remote denial of service. The issue is triggered when handling shun events, and will result in loss of availability for the program via specially crafted IP packets.

Related Work

- Bessani et al., *The CRUTIAL Way of Critical Infrastructure Protection*, IEEE Security and Privacy, 2008
- Roeder & Schneider, *Proactive Obfuscation*, ACM Transactions on Computer Systems, 2010

- Some of the limitations of the previous solution
  - used **simple filtering** mechanisms
  - required **specific communication hardware** support
  - imposed **strict timeliness** assumptions in part of the system
  - relatively **static configuration** of the replicas



Support both stateless and stateful application level filtering

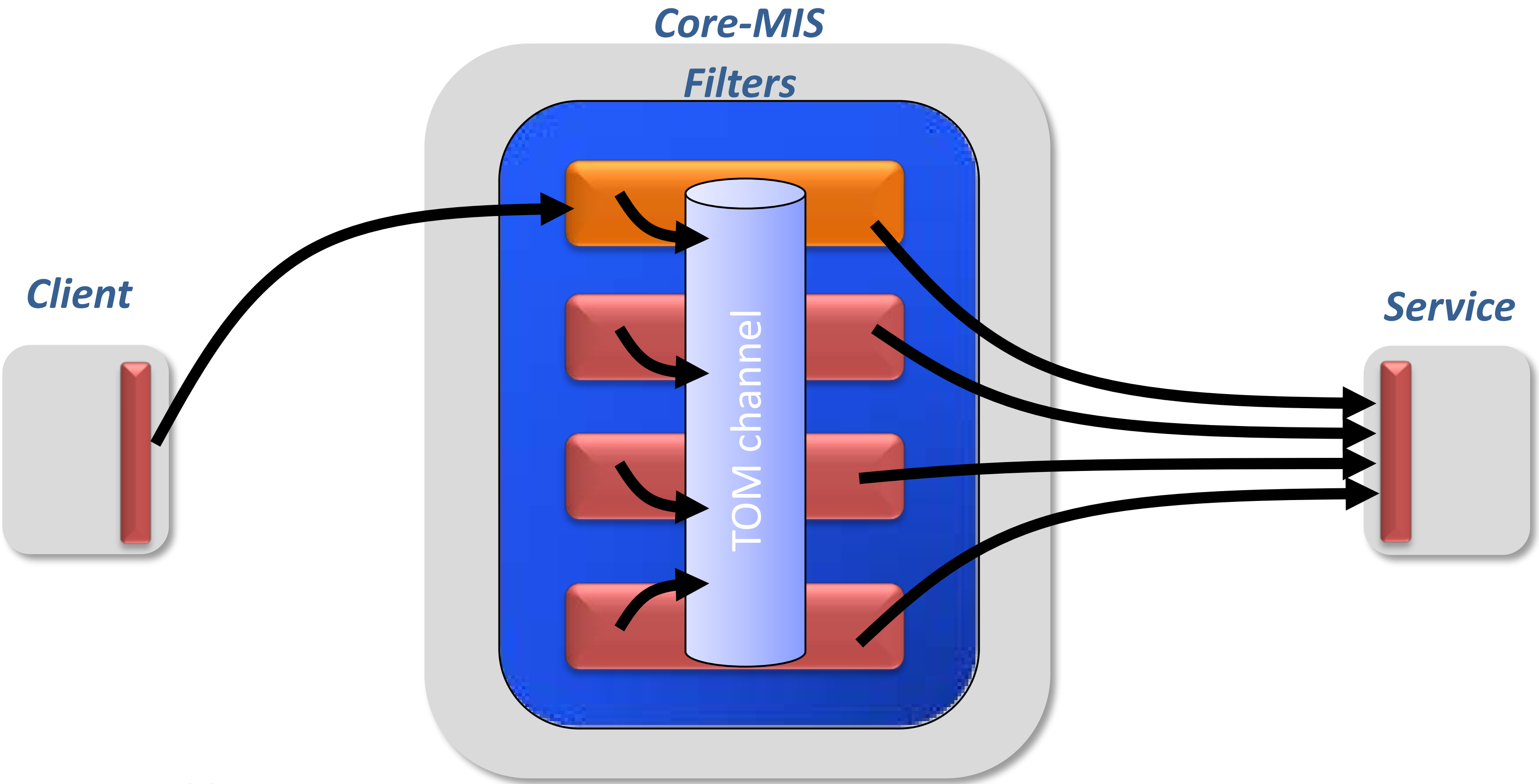
Eliminate need for specific hardware

Reduce synchrony assumptions

Adapt to failure conditions

OUR GOALS VERSUS RELATED WORK

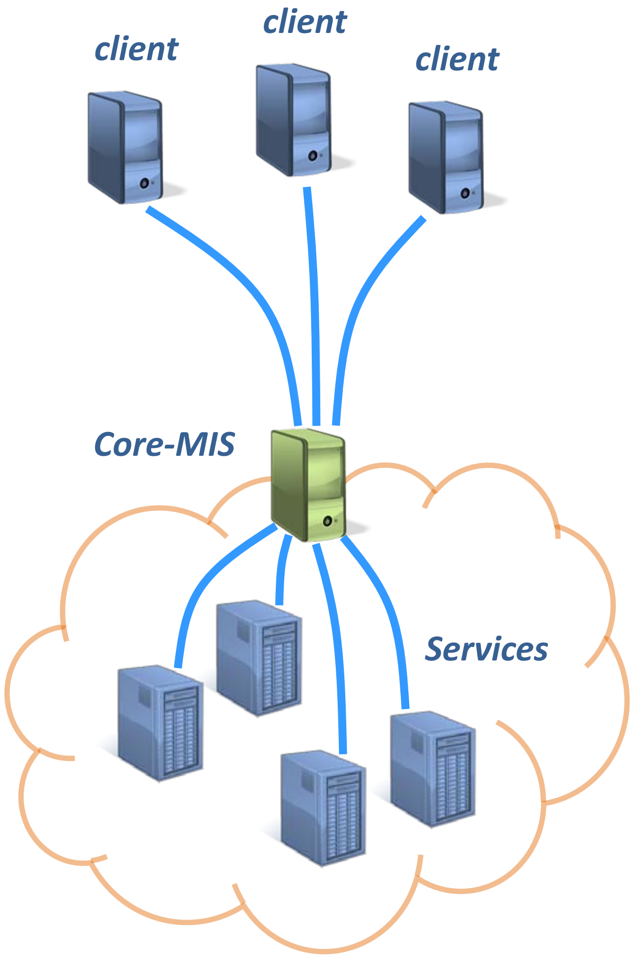
Design based on Related Work



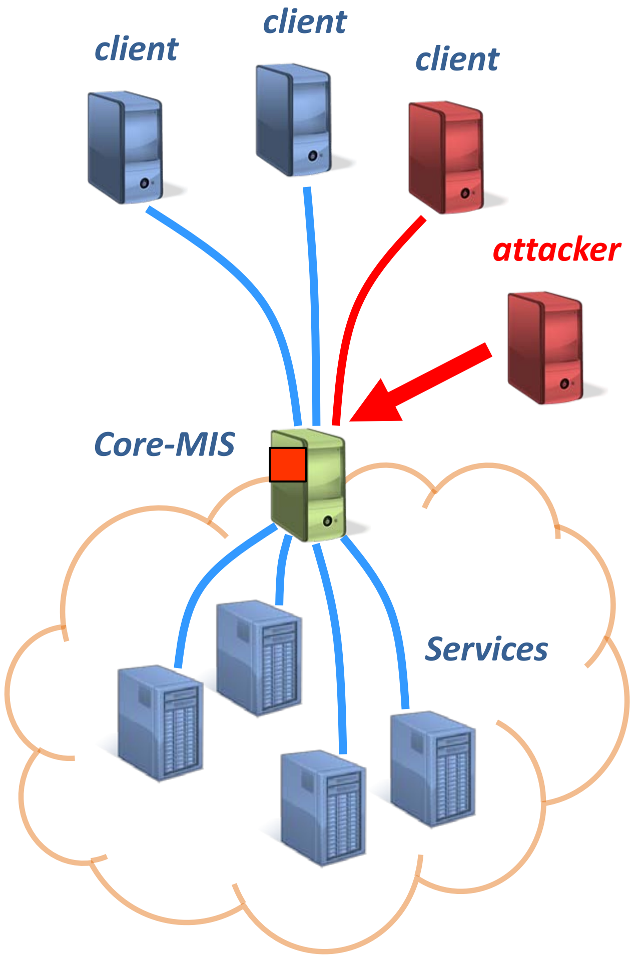
Potential limitations:

- ***a primary failure (detection, new leader, recover consistent state)***
- ***load is not distributed evenly (primary is more loaded)***
- ***filters had to separate good and discard bad traffic***
- ***the number of filters is typically static***

Overview

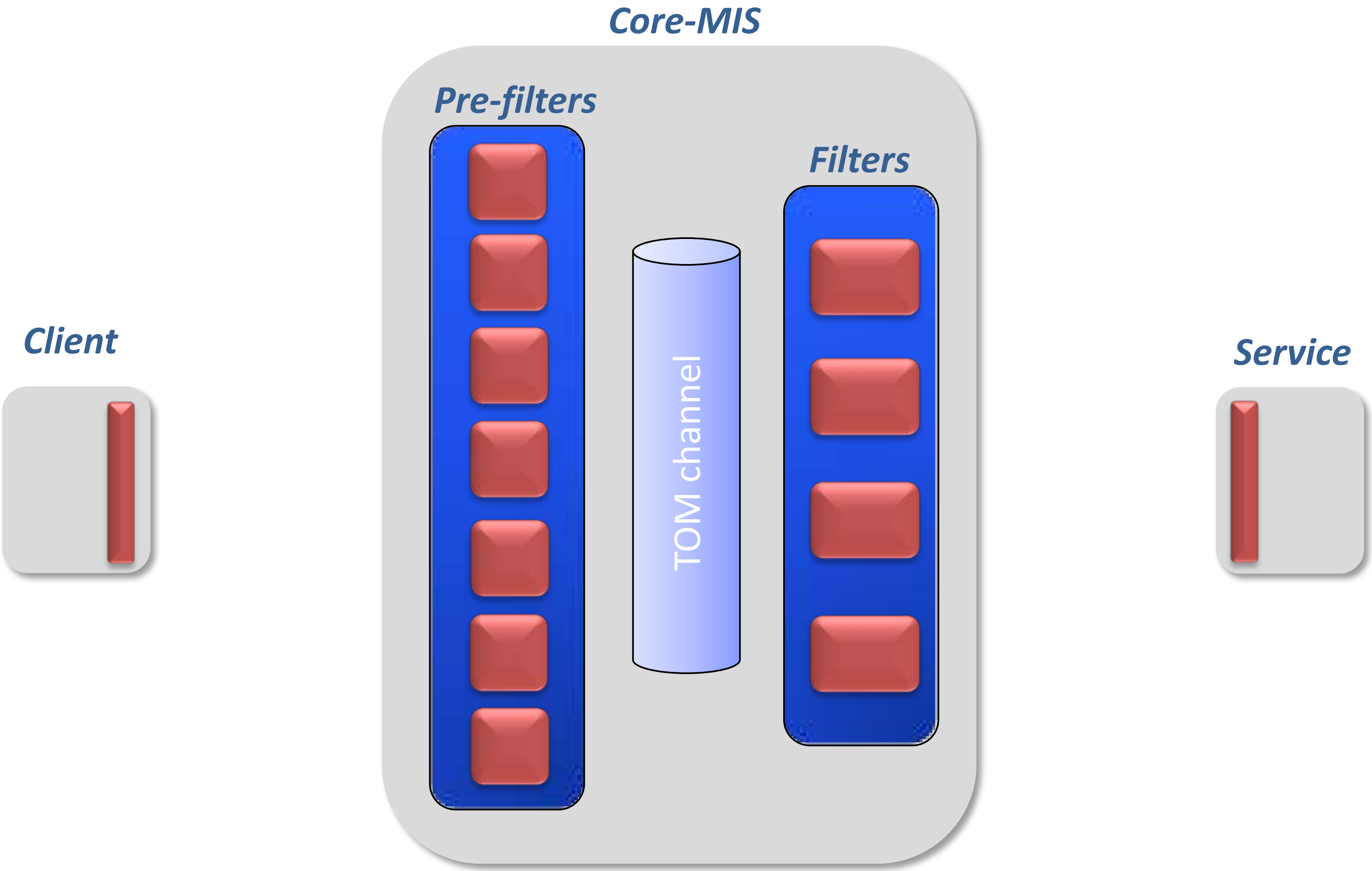


Overview



- Wish list
  - ensure that only benign traffic goes through with high probability
  - highly resilient to a variety of failures
    - crashes & intrusions
    - DoS & other network attacks
  - good performance
- Willing to tradeoff
  - some level of transparency

Design

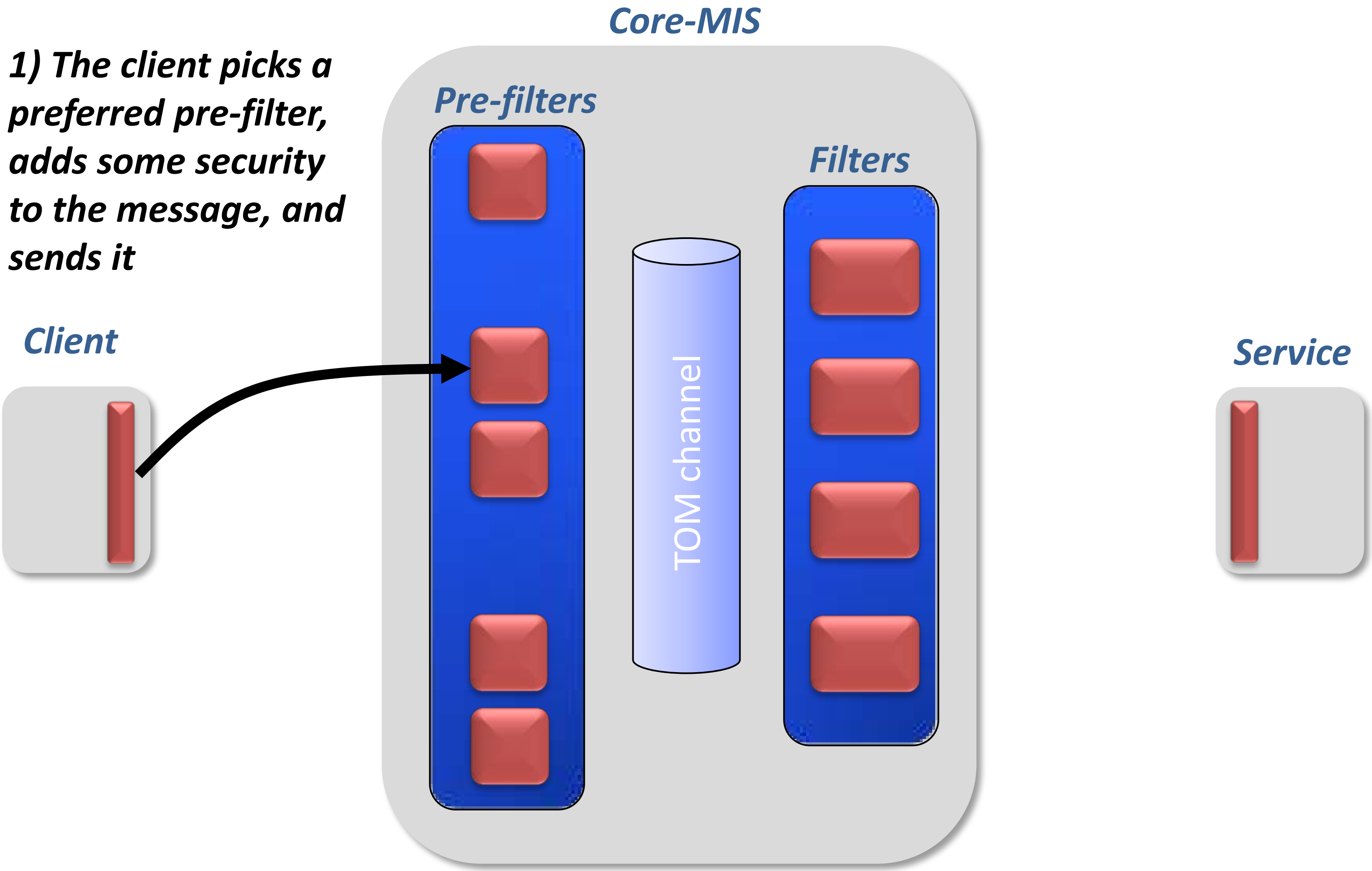


Main Components

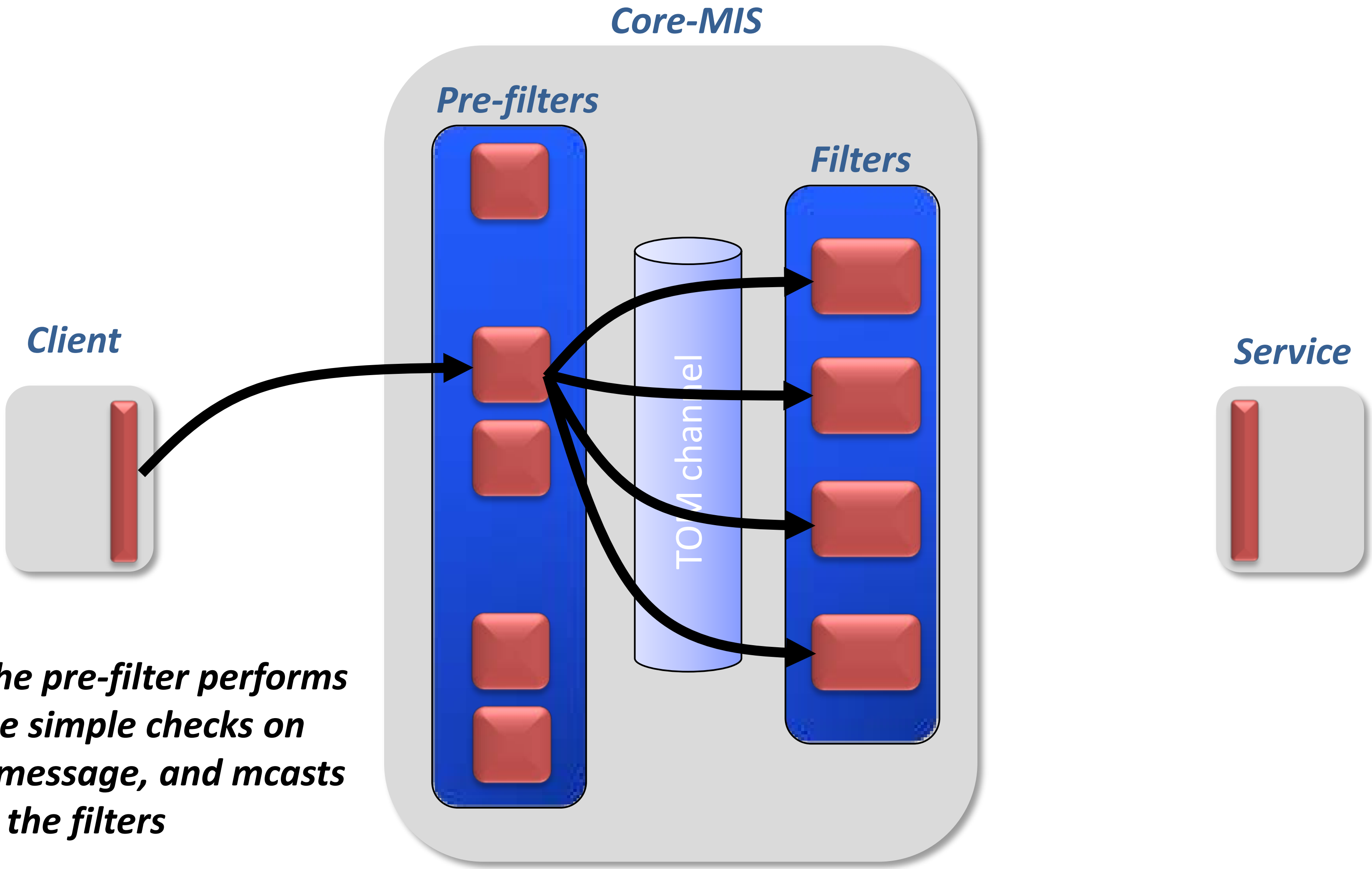
- Pre-Filters
  - lightweight, simple filtering operations to be efficient
  - are created/removed **dynamically** to adapt to client load & respond to DoS attacks and intrusions
  - employ **detection & recovery** techniques for resilience
  - tolerates up to *m-1* failures out of *m* pre-filters
- TOM Channel
  - asynchronous Byzantine total order multicast
- Filters
  - application level security policies and filtering
  - (mainly) static group
  - uses **masking techniques** based BFT replication
  - tolerates up to (n-1)/3 failures out of n filters

Normal Operation

1) The client picks a preferred pre-filter, adds some security to the message, and sends it

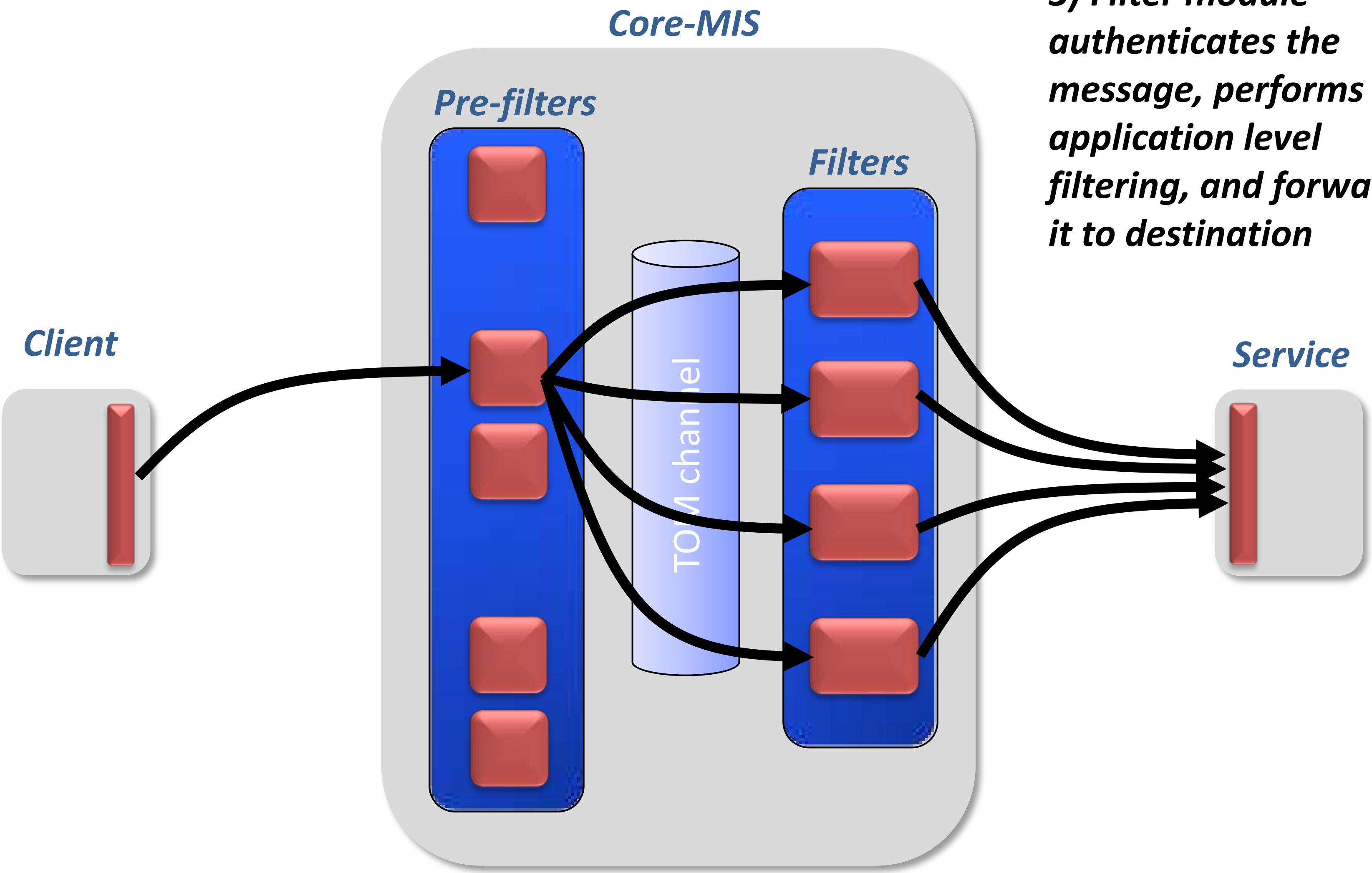


Normal Operation



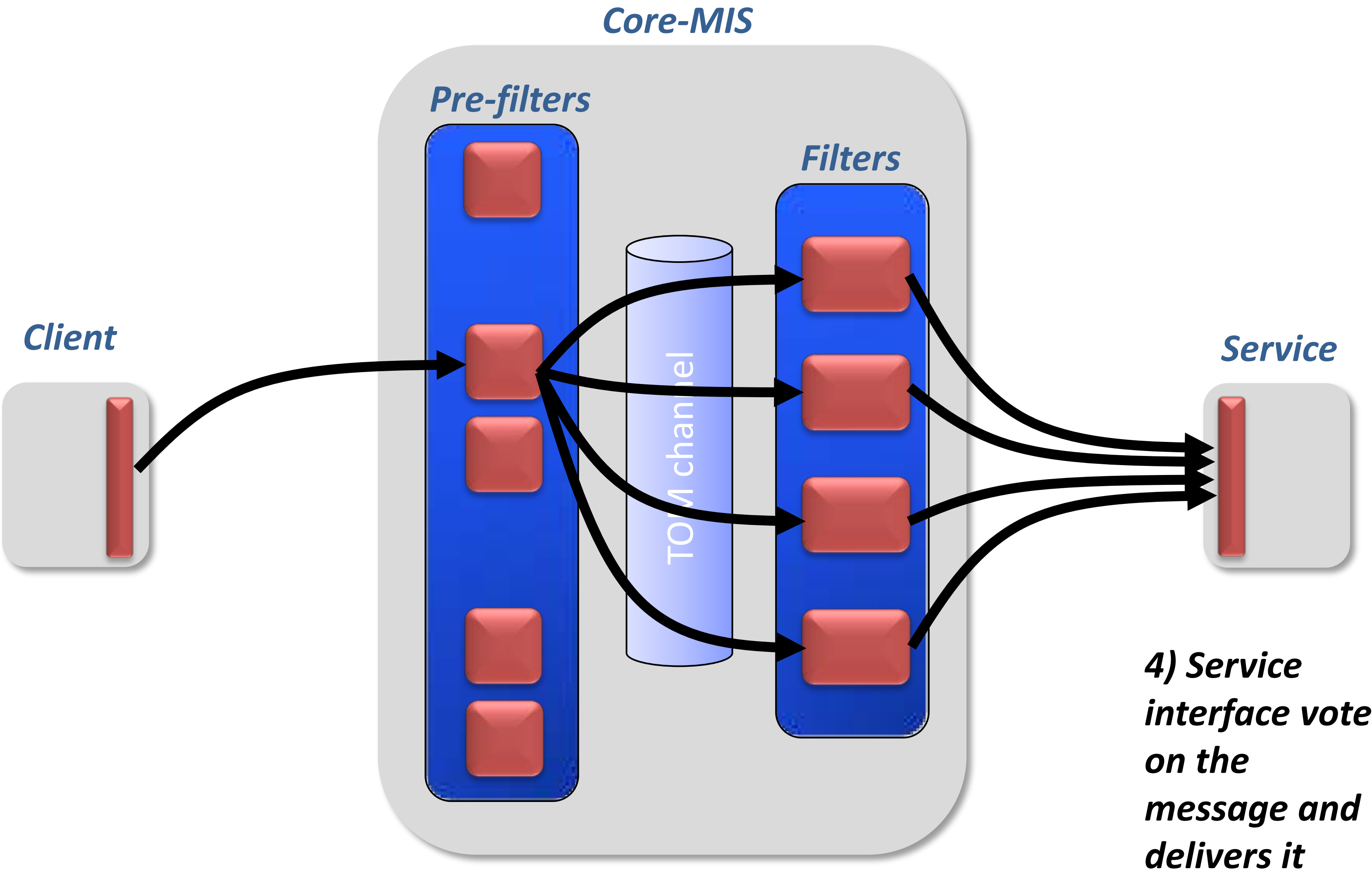
2) The pre-filter performs some simple checks on the message, and mcasts it to the filters

Normal Operation



3) Filter module authenticates the message, performs application level filtering, and forwards it to destination

Normal Operation



4) Service interface votes on the message and delivers it

Mechanisms for Addressing Failures

- Outside attackers

  [Integrity/Auth] message authentication (i.e., MACs) at clients

  [Availability/DoS]

  i) discard malicious traffic as fast as possible;

  ii) replace the pre-filters under attack with new ones

  Limitation: attacker completely fills the channels to the core-MIS

- Malicious clients

  [Integrity/Auth] perform application filtering

  [Availability/DoS] impose throttling limits and discard extra traffic

  Limitation: can not detect all malicious content

Mechanisms for Addressing Failures (cont.)

- Compromised pre-filters

  [Integrity/Auth] cannot fake authentication

  [Availability/DoS] replace misbehaving pre-filters

- Compromised filters

  [Integrity/Auth] voting at the final service

  [Availability/DoS] *eventually use proactive-reactive recovery*

Thank you for your attention!

Questions?