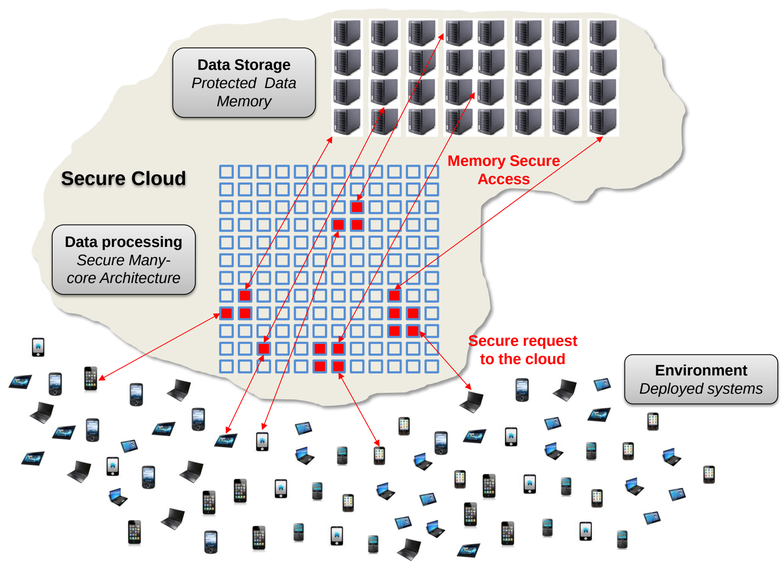
Secure Deployment in trusted Many-core Architectures

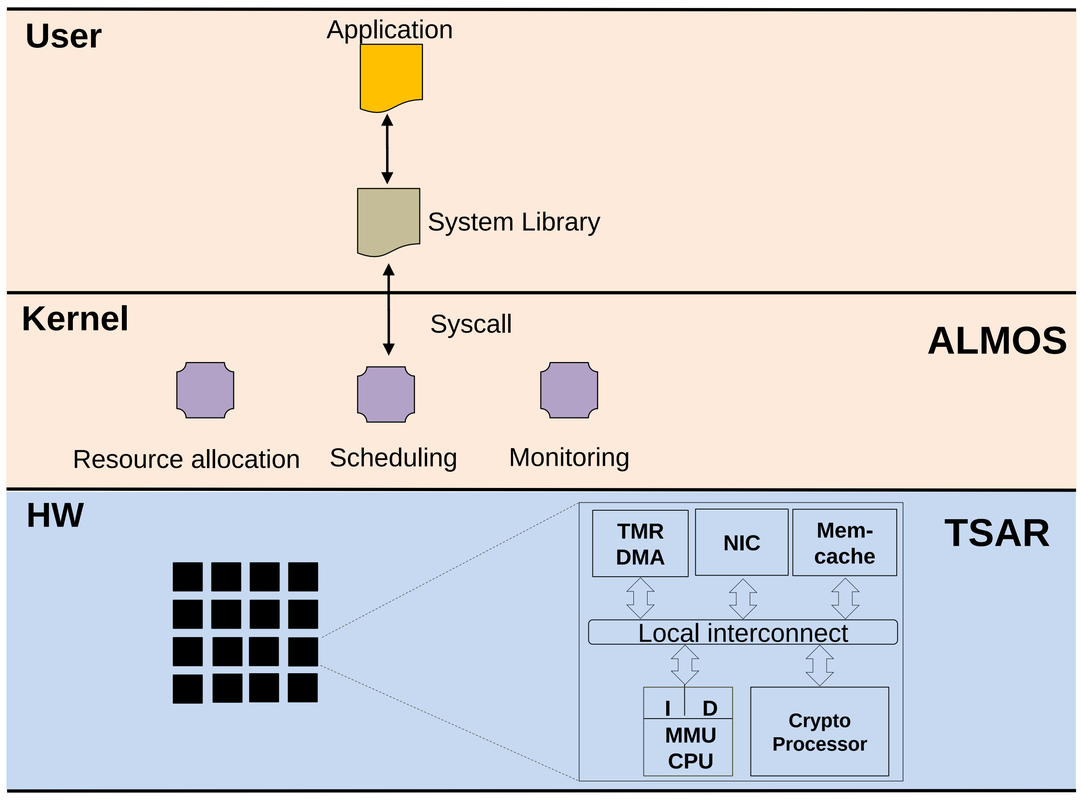
Maria Méndez Real*, Cuauhtémoc Mancillas Lopez¥ ,Guy Gogniat*, Lilian Bossuet ¥,
Adel Baganne*, Victor Fischer ¥,

*Univ. Bretagne Sud, Lab-STICC, firstname.lastname@univ-ubs.fr
¥ Université de Lyon, Laboratoire Hubert Curien, firstname.lastname@univ-st-etienne.fr

TSUNAMY ANR Project (2013-2017)

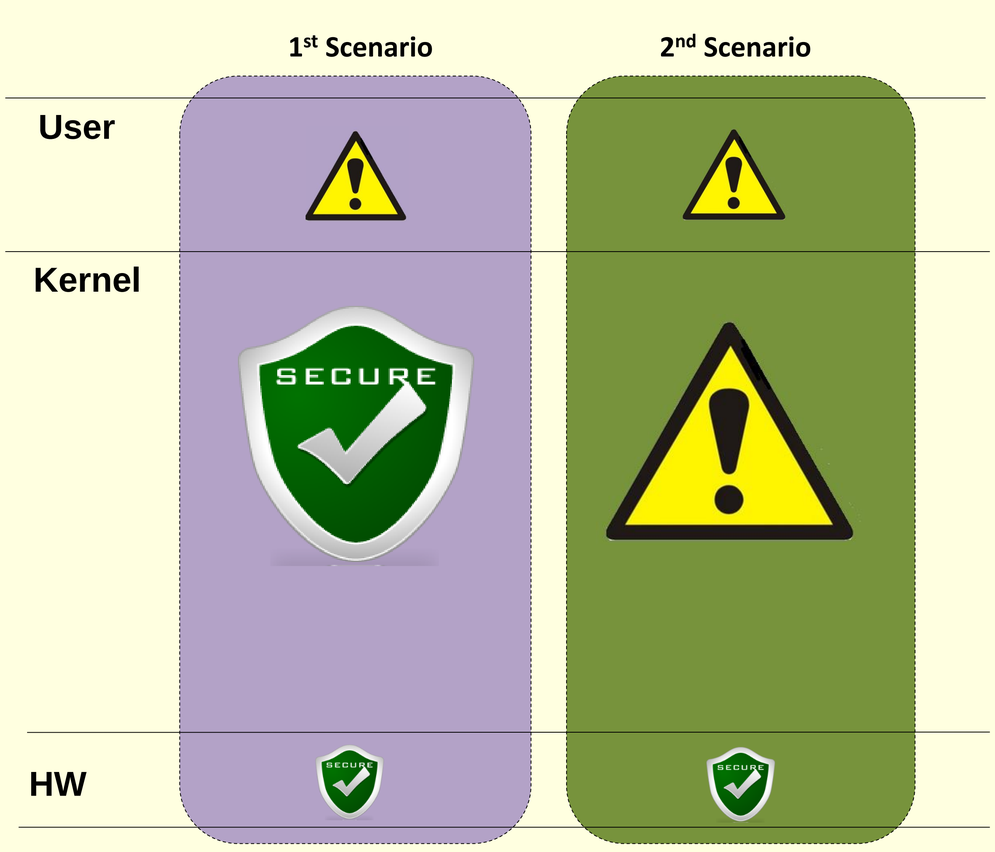


Many-core Architecture (up to 1024 cores)



Many-core Architecture Threats Model

Different scenarios



Threats model

Mean of attack	Threat	Risk level
1 Malicious process	Denial of services	High
	Information leakage	Medium
N Malicious processes	Denial of service of the targeted peripheral	High
Malicious monitoring	Denial of services	Low
Malicious allocation	Unauthorized read of data in memory	Low/Medium
	Denial of services	Low/Medium
Malicious control of memory access rights	Denial of services	Low/Medium
	Information leakage	Low/Medium
Malicious scheduler	Denial of service	Low/Medium
	Information leakage	Low/Medium
Malicious peripheral driver	Unauthorized read of data in memory	Medium
	Unauthorized write of data in memory	Medium
	Denial of services	Medium
Malicious programmable interrupt controller	Denial of services	Low/Medium

Trusted ALMOS: Secure Application Deployment

Related work *

Secure Services	Secure Deployment Properties		
	Main function	Potential Attack	Required Information
Scheduling and resources allocation	Scheduling	Denial of services	Scheduling policy priorities
	Task placement	Information leakage	Application sensitivity, resources needs and communication Global system state
	Dynamic resources allocation	Unauthorized read of data in memory Denial of services	
Control	Control of maximal resources utilization	Denial of services	Maximum CPU and crypto processor Utilization time
Security	Context awareness	Unauthorized read of data in memory	Application resources needs and communication
	Reset resources after use	Information leakage	-
	Protect communications between sensitive and non-sensitive applications	Information leakage	Sensitivity and task communication
	Securely sharing crypto processor key	Unauthorized read of data in memory	

OS services needed to be secure: Scheduling, Resource allocation, Monitoring

Perspectives

- **ALMOS extension in order to guarantee a trusted execution of parallel applications**
  - Theoretical approach through system modeling (application, architecture and deployment algorithm)
  - SystemC simulation of the complete system (ALMOS and TSAR extended with cryptoprocessors)
- **Software and hardware mechanisms to guarantee security policies of applications**
  - Software level: Secure services within ALMOS
  - Hardware level: Firewall to filter unauthorized accesses

* R. J. Masti *et al.* (2012). Enabling Trusted Scheduling in Embedded Systems. *Proceedings of the 28th Annual Computer Security Applications Conference*, 61-70

R. J. Masti *et al.* (2014). Isolated Execution in Many-core Architectures. *Network and Distributed System Security Symposium*.