

The logo for INAETICS is a stylized gear or circular structure composed of several interlocking segments in various colors including orange, red, yellow, green, and blue. It is positioned on the right side of the slide, partially overlapping the title and date.

A software architecture for high-availability systems

October 1st, 2015

Hans Bossenbroek (hans.bossenbroek@luminis.eu)
René van Hees (rene.vanhees@nl.thalesgroup.com)

www.inaetics.org

INAETICS

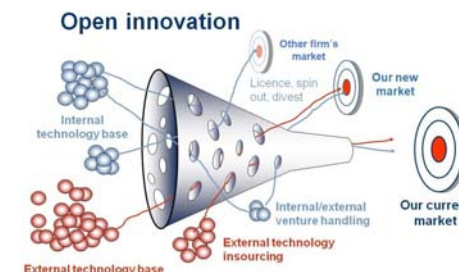
The INAETICS mission



INtelligent robust ArchitecturE for Time Critical Systems

“ An *open collaboration effort* that aims to define and demonstrate a *dynamic service oriented reference architecture* that addresses the requirements of *time critical systems* in a *broad range of domains* by providing a *single design and implementation space* for all subsystems, *irrespective of control strategy*. ”

The Consortium:

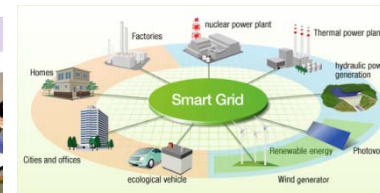


Agenda



1. Introducing INAETICS
2. Architectural principles
3. Architecture overview
4. Core Architectural Mechanisms
 1. Component Service Model
 2. Security Model
 3. Coordination Model
5. Roadmap

The INAETICS domains



Defense
market

Personalized
Learning

Smart
Grids

INAETICS Architecture

The realization is directly driven by the high-availability, deterministic and geographically dispersed character of the domains that are targeted by the INAETICS project

Software evolution?



Software is subject to change in the course of its existence

Lehman's Laws of software evolution (1974 -)

E-systems (systems with real-world behavior):

- Continuing Change
- Increasing Complexity
- Fundamental Law of Program Evolution
- Conservation of Organizational Stability
- Conservation of Familiarity

Software evolution is not Darwinian, Lamarckian or Baldwinian,
but an important phenomenon on its own.

Architectural principles



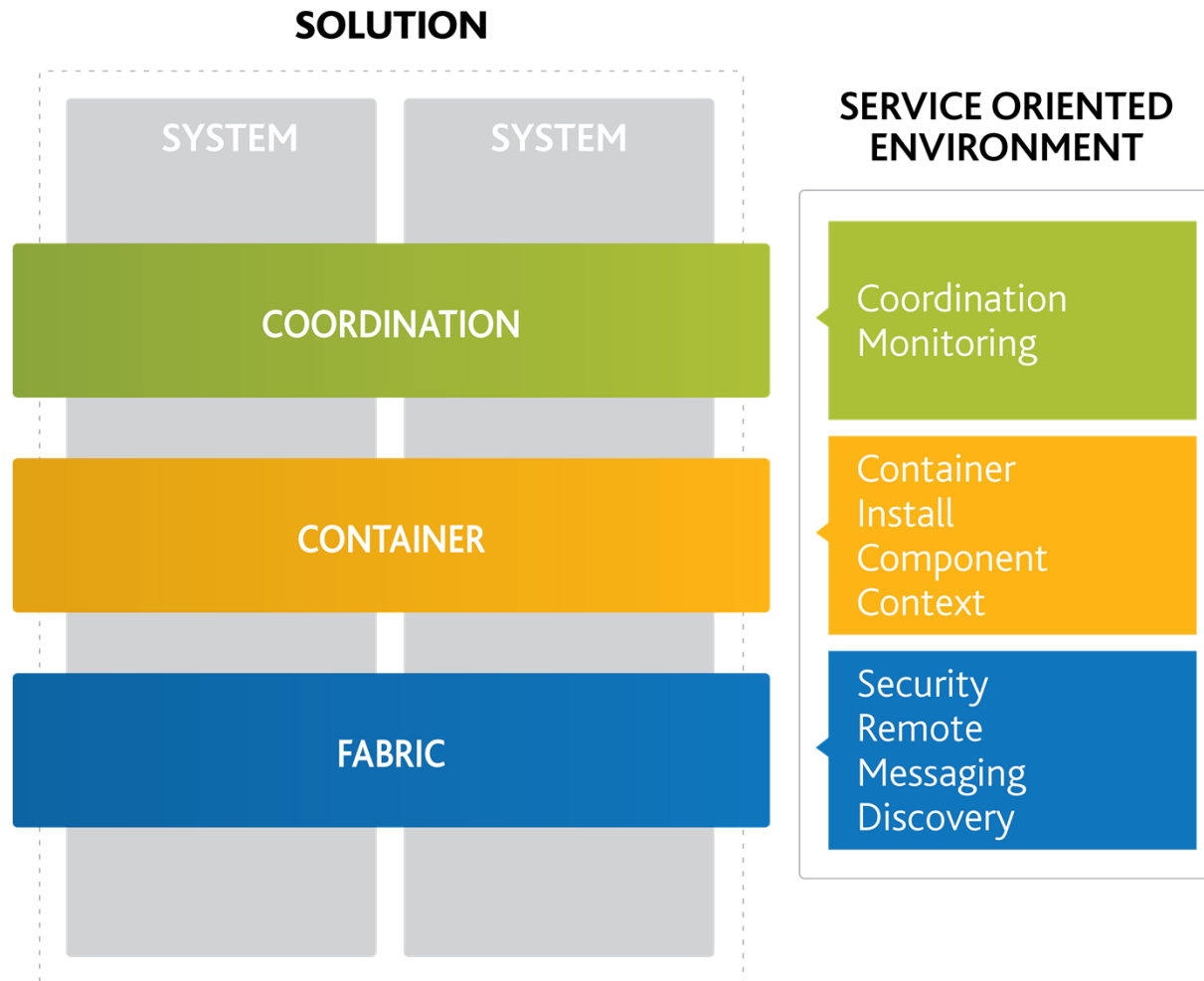
Evolution must be designed into systems

How to design evolvability into complex, distributed systems?

- Software Modularity
- Dynamic Component-Services Architecture
- Dynamic application assembly and deployment
- Risk-adaptive security architecture

Applying compositional techniques with a design that is ultimately controlled by a dynamic coordination strategy

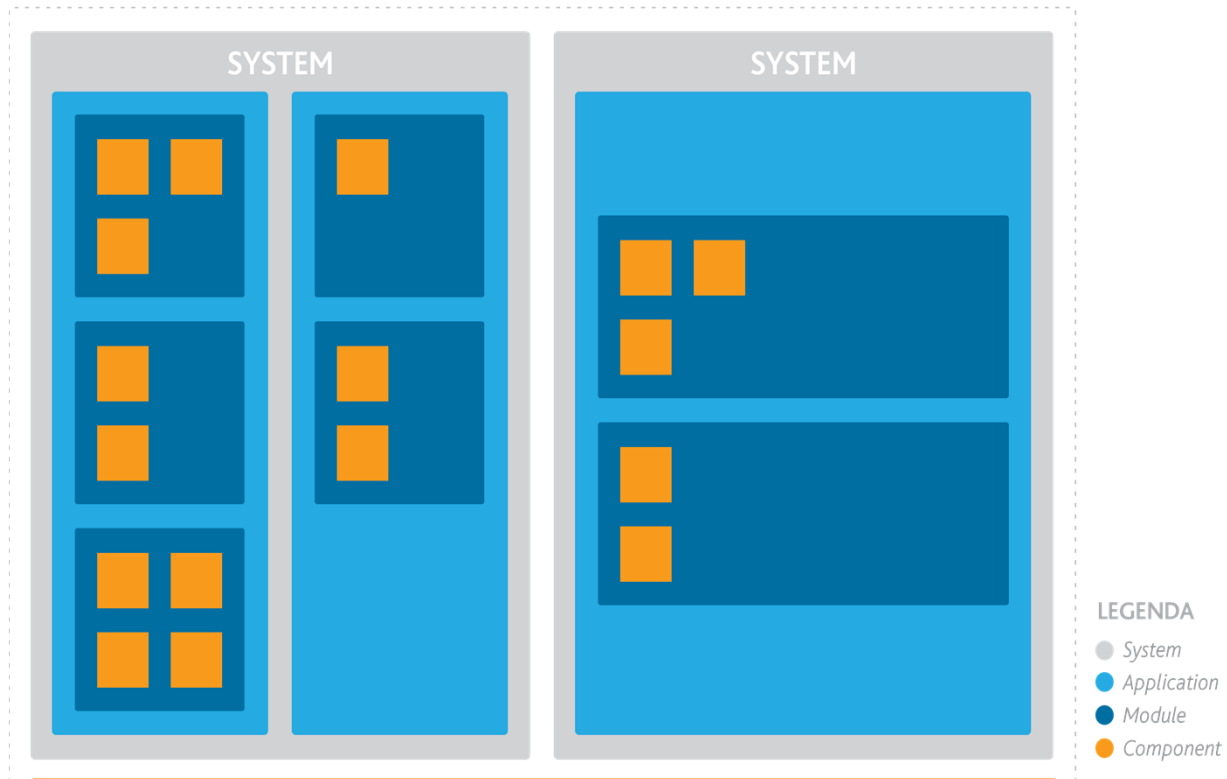
The INAETICS layering



Component – Service Model



SOLUTION



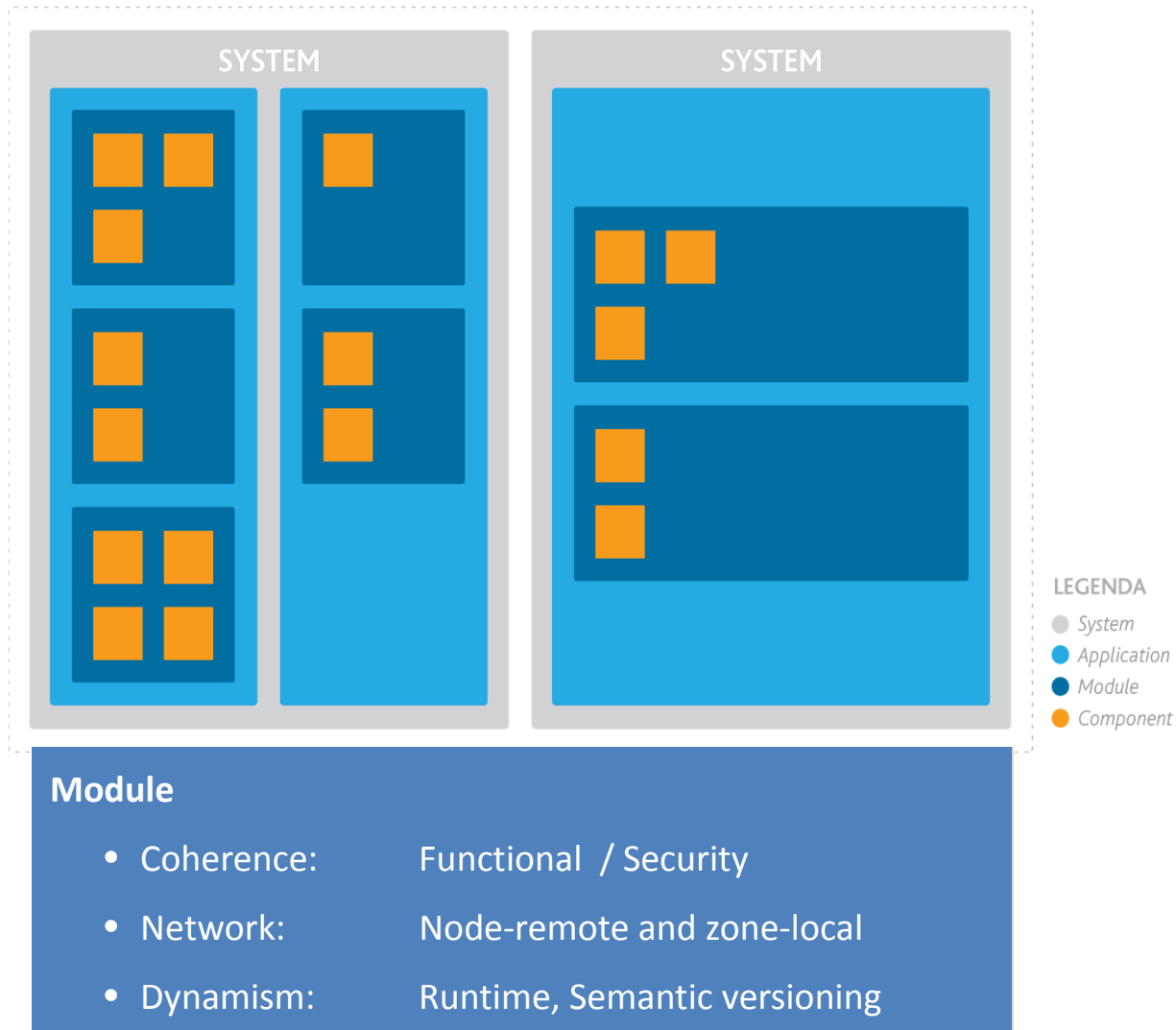
Component

- Coherence: Physical
- Network: Node-local
- Dynamism: Code-time

Component – Service Model



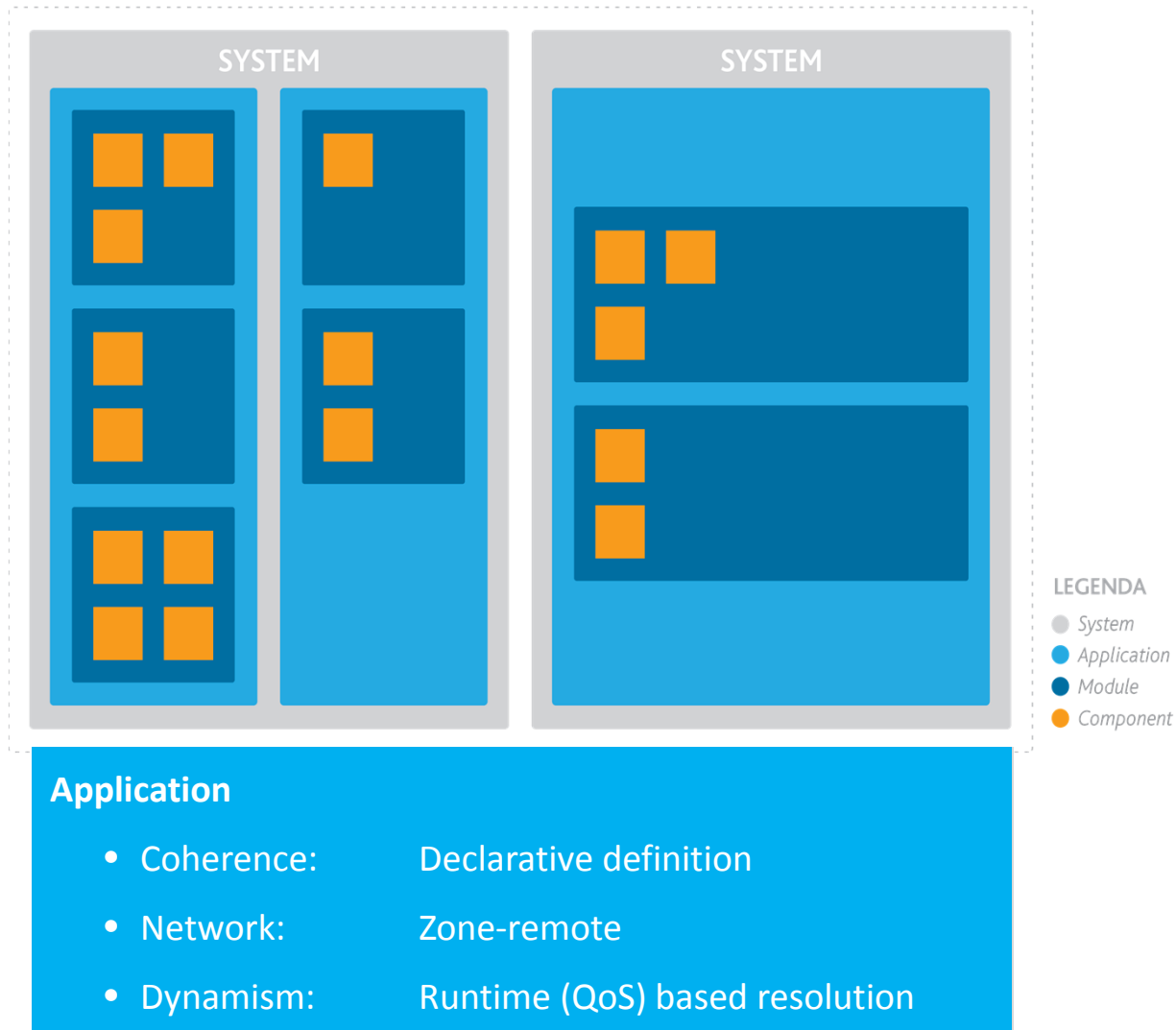
SOLUTION



Component – Service Model



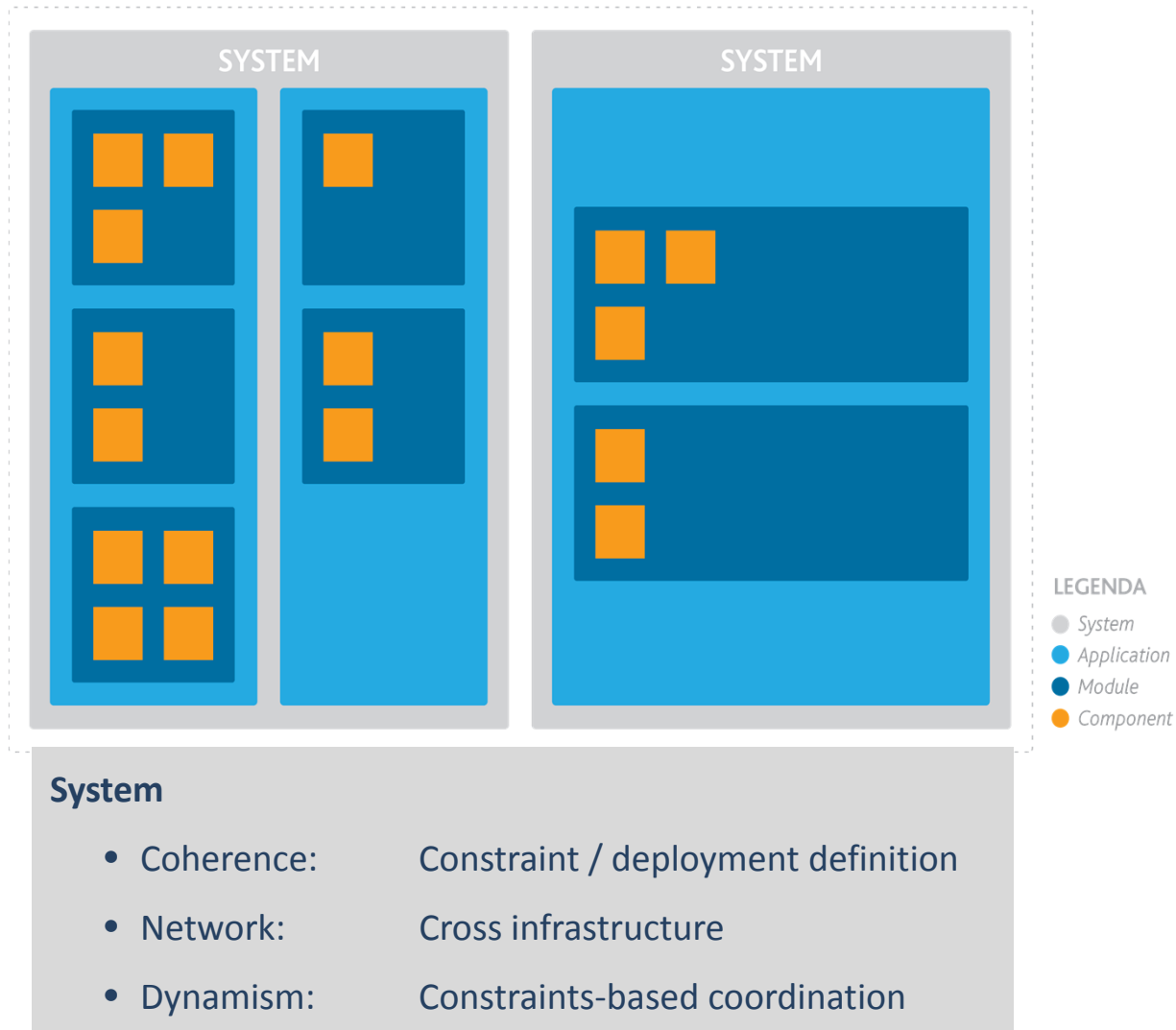
SOLUTION



Component – Service Model



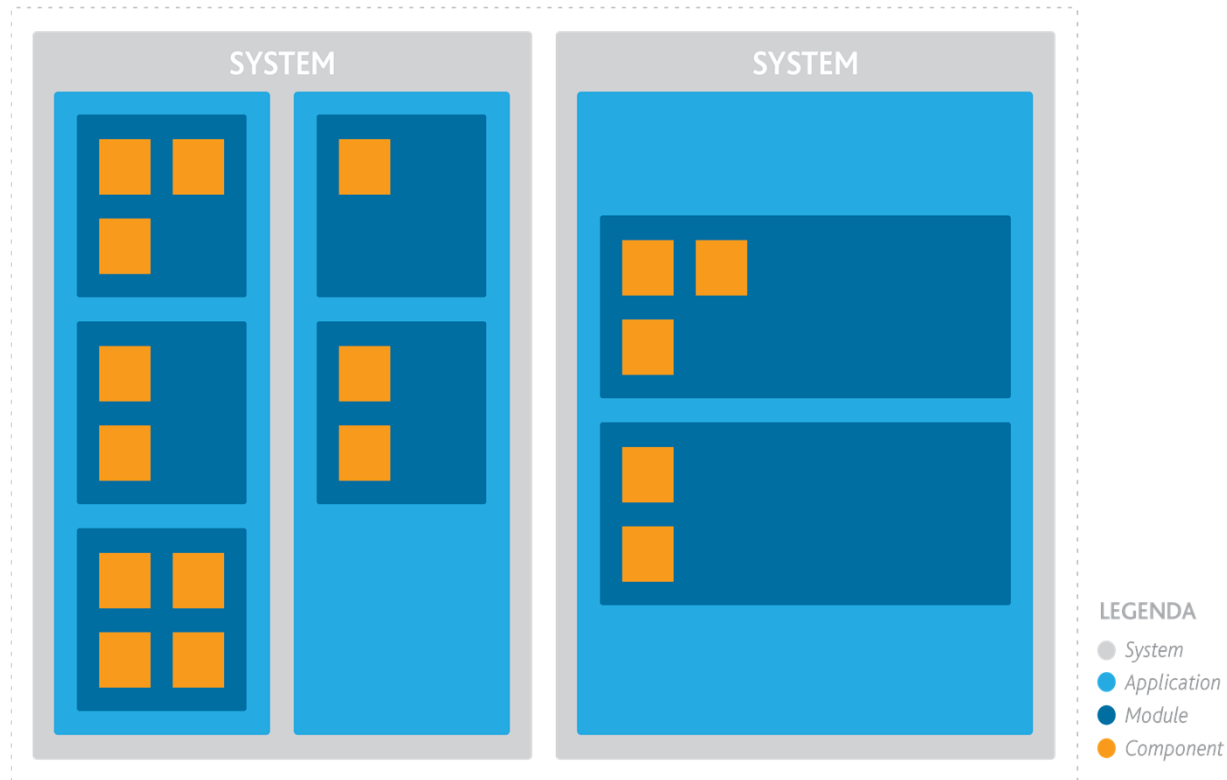
SOLUTION



Component – Service Model



SOLUTION



Solution

- Coherence: Intent definition
- Network: Infrastructure neutral
- Dynamism: Intent-based coordination

Security Model



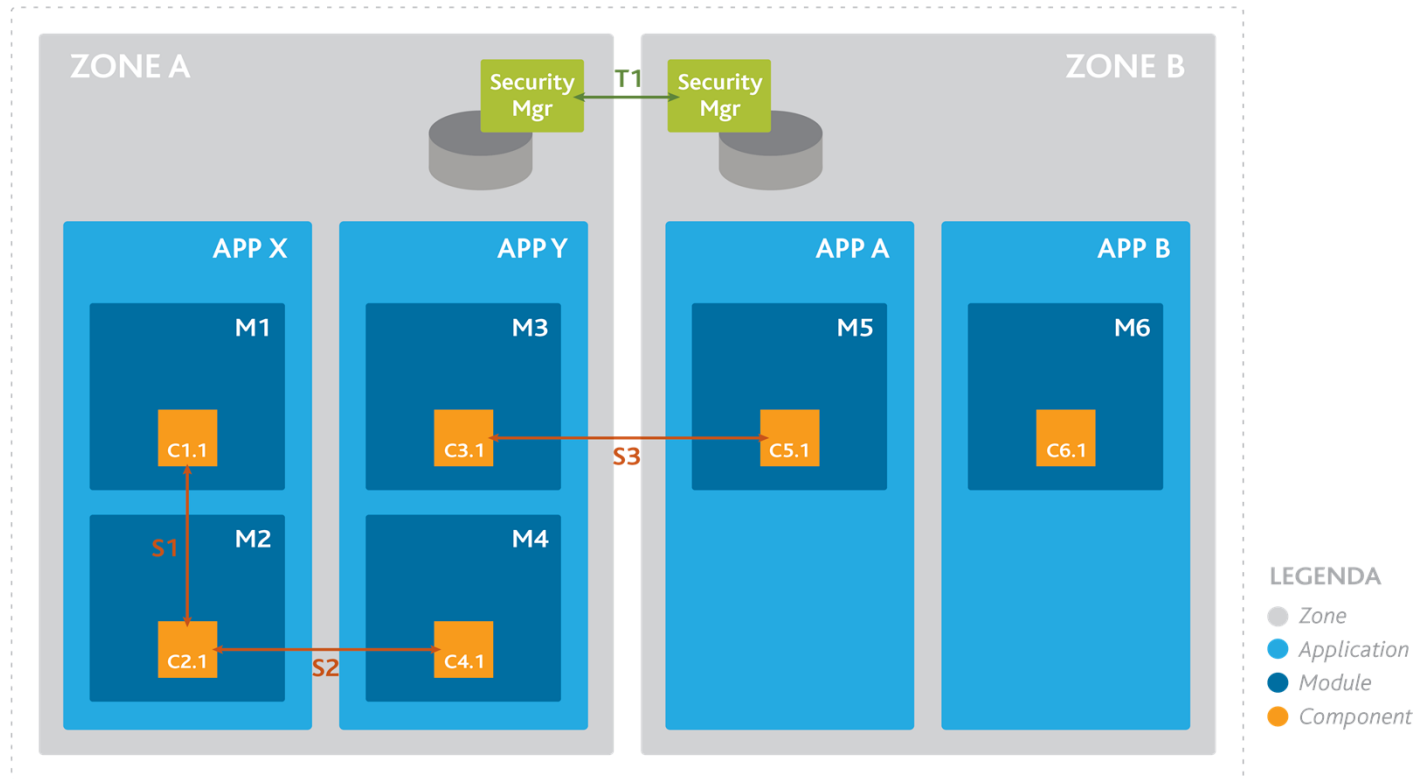
Is based on:

- No introduction of new structural parts to the Fabric layer
- A zone is the basic security concern
- Trusted connections and encrypted communication across multiple zones can only be attained at module level
- There is always a security manager responsible for defining and enforcing policies

Security Model



SOLUTION



Security Model

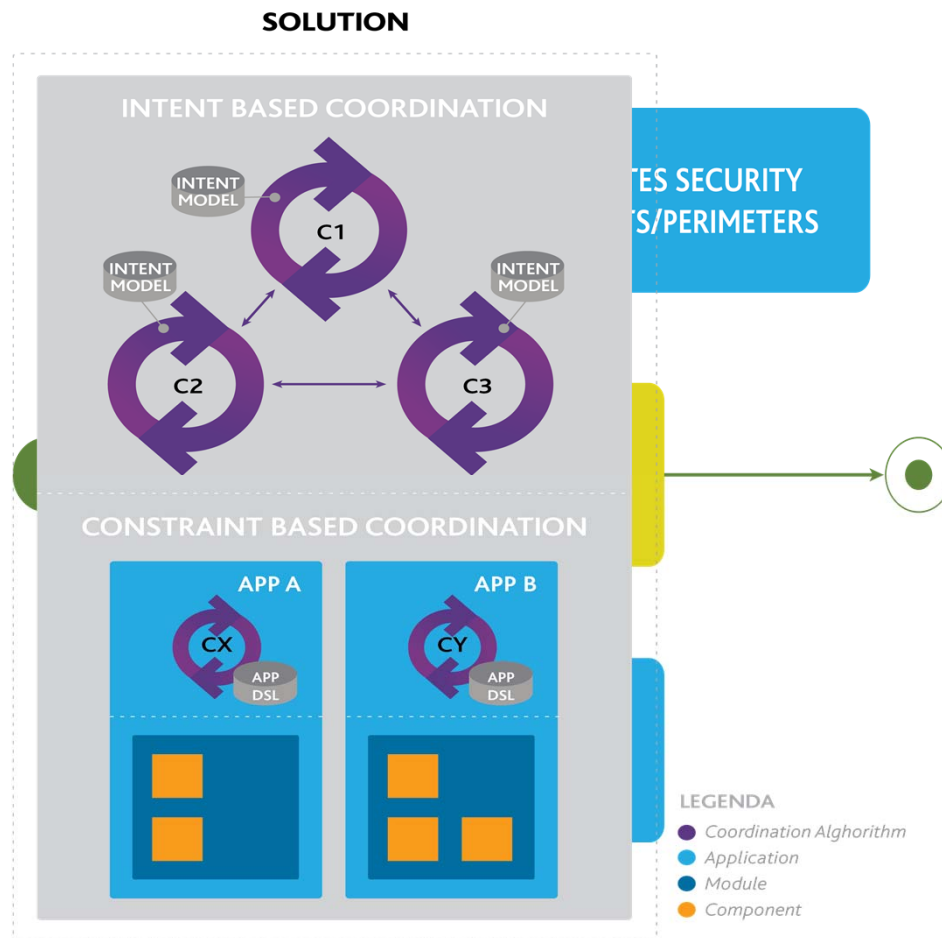


Includes:

- Attribute based encryption
- Short-lived trust relations
- Non centralized security management

to enable to dynamically reestablish trust-relations and security perimeters using INAETICS coordination, with a minimum of human intervention.

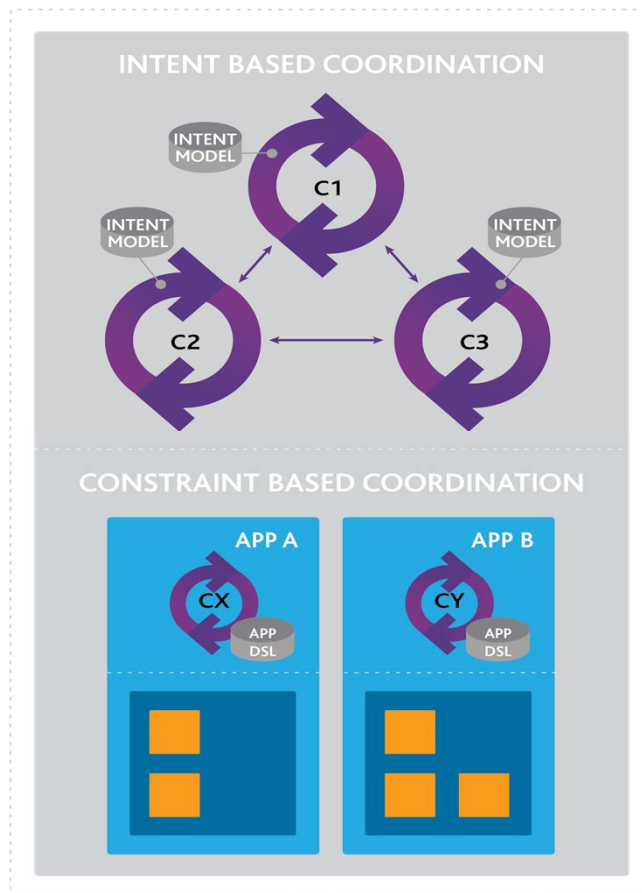
Coordination Model



Coordination Model

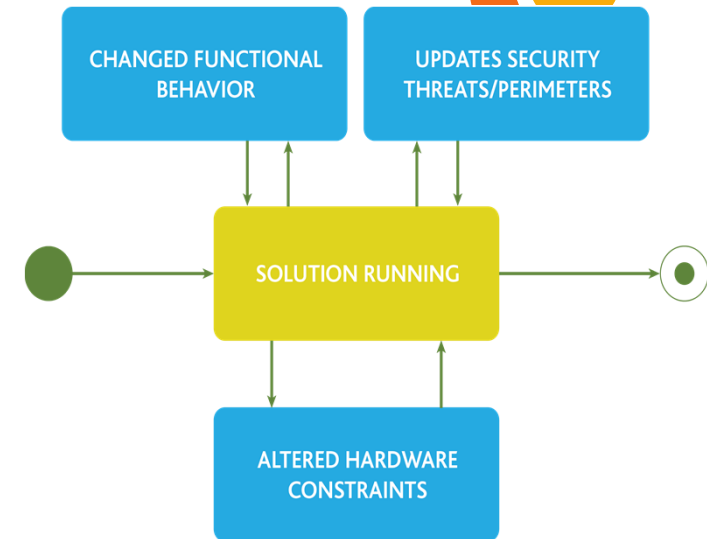


SOLUTION



LEGENDA

- Coordination Algorithm
- Application
- Module
- Component

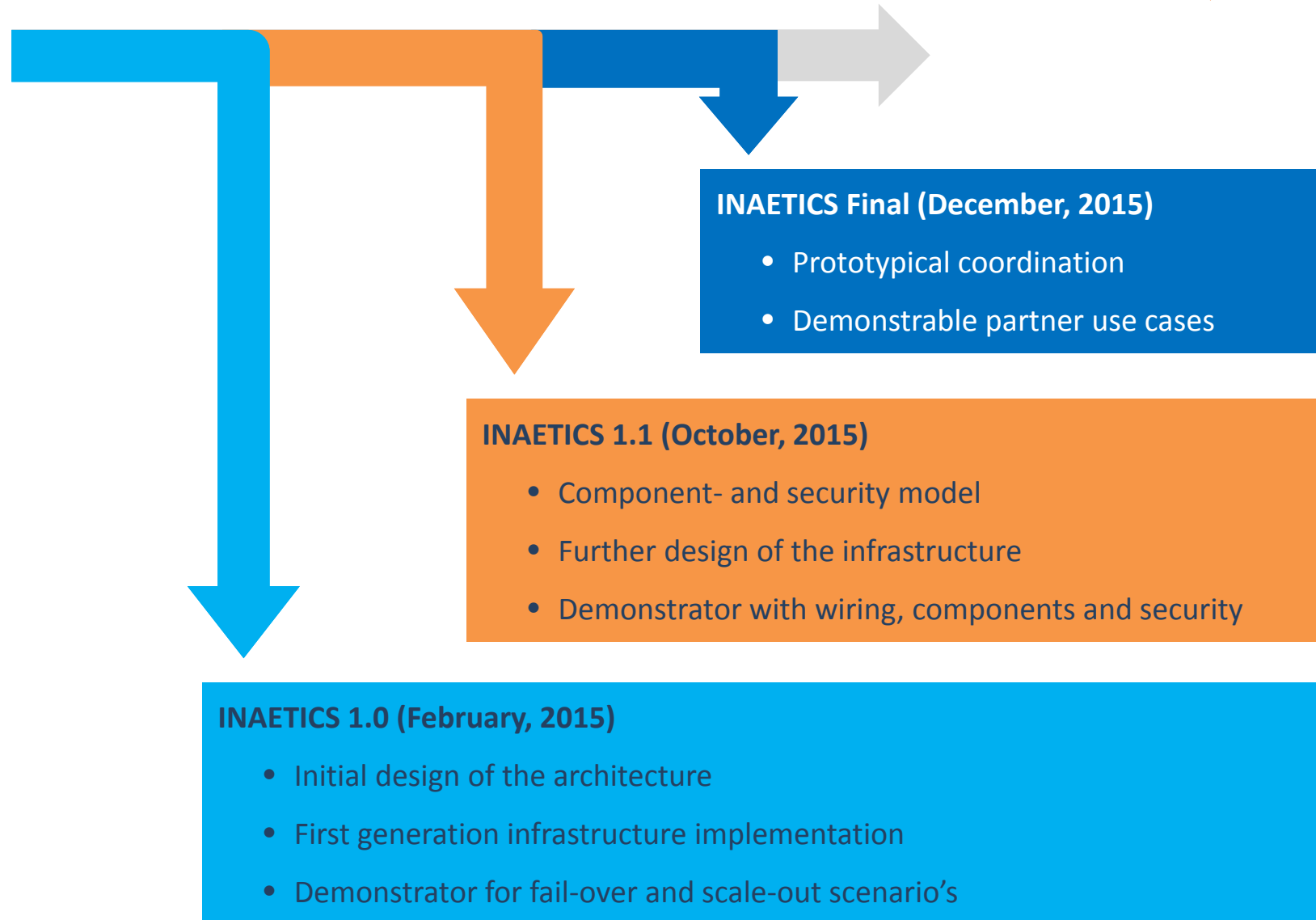


The INAETICS roadmap



March, 2013

December, 2015



More INAETICS information



Software Architecture evolution in an Open World *Introducing the INAETICS project*



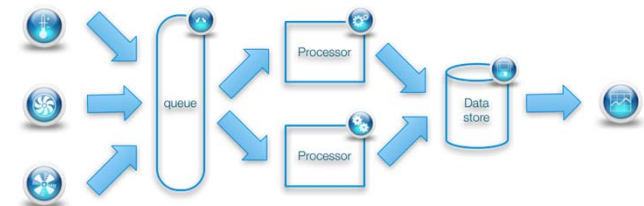
Auteurs
Hans Bossenbroek
René van Hees



The INAETICS architecture *Introducing INAETICS*



Authors:
Hans Bossenbroek
René van Hees



INAETICS 1.1

- Component- and security model
- Further design of the infrastructure
- Demonstrator with wiring, components and security

@ www.inaetics.org



More INAETICS information



**Software Architecture
evolution in an Open World**
Introducing the INAETICS project

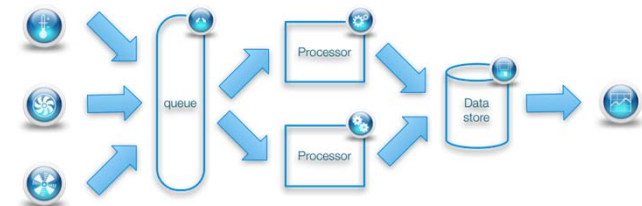


Auteurs
Hans Bossenbroek
René van Hees

The INAETICS architecture
Introducing INAETICS

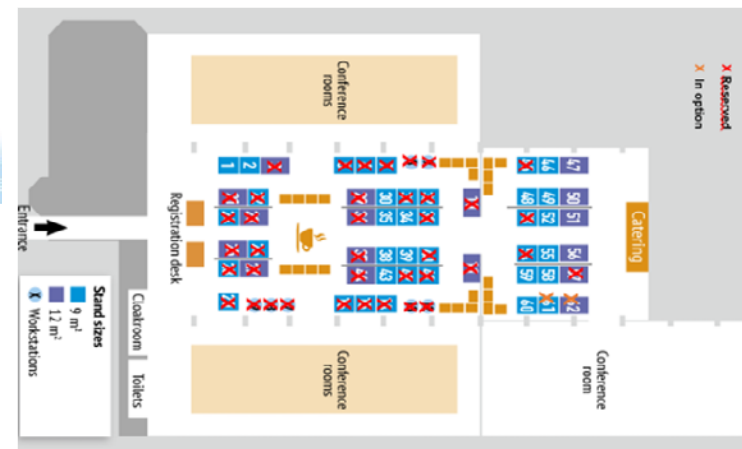


Authors:
Hans Bossenbroek
René van Hees



INAETICS 1.1

- Component- and security model
- Further design of the infrastructure
- Demonstrator with wiring, components and security



More INAETICS information



Software Architecture evolution in an Open World *Introducing the INAETICS project*

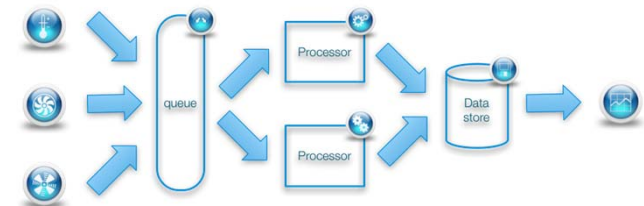


Auteurs
Hans Bossenbroek
René van Hees

The INAETICS architecture *Introducing INAETICS*



Authors:
Hans Bossenbroek
René van Hees



INAETICS 1.1

- Component- and security model
- Further design of the infrastructure
- Demonstrator with wiring, components and security



11:30 – 12:00

How to implement a robust software architecture using open source solutions

Jan Willen Janssen, Luminis & Gerrit Binnenmars, Thales

How do you build an architecture based on open source components that is robust, scalable, secure and dynamic that isn't outdated right away when put into production while still being applicable for different use cases? The Inaetics project will demonstrate how its architecture achieves this.

