THALES

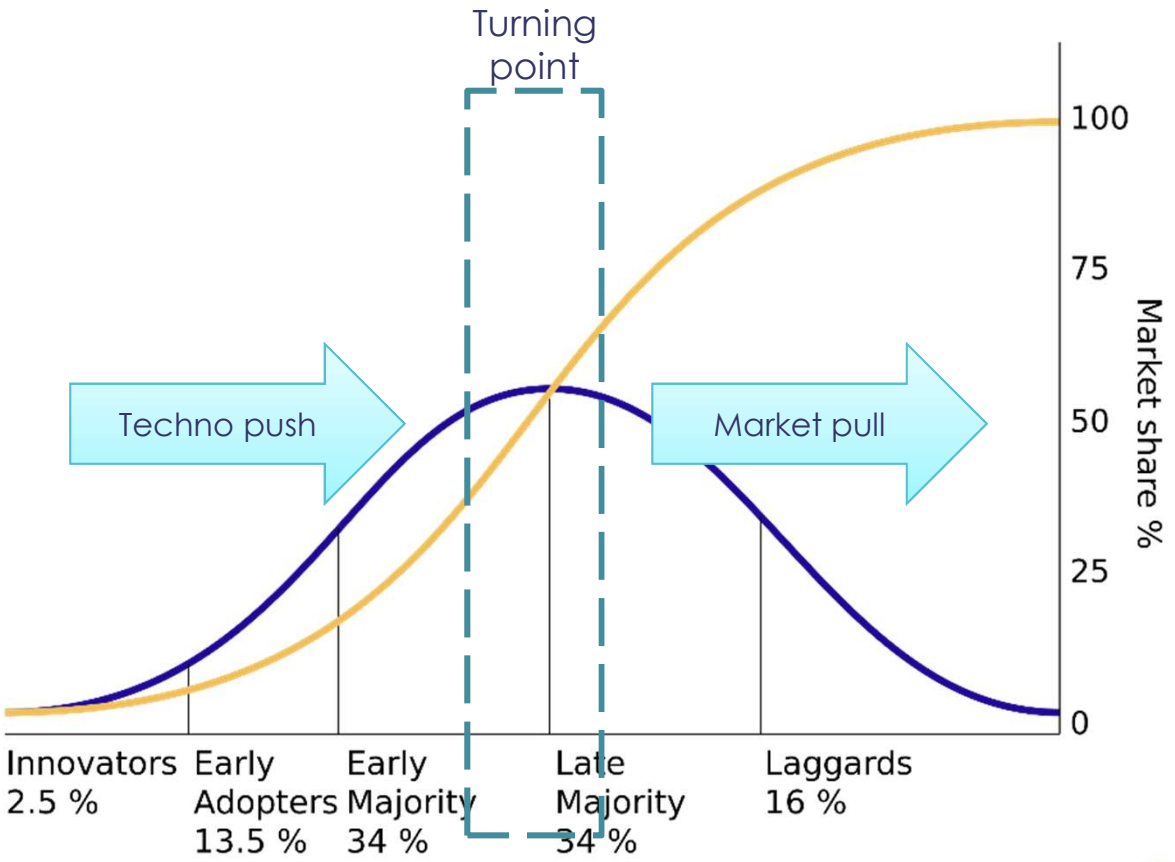
Resilience in evolving software systems

CIGRE B5

René van Hees
Chief software architect @Thales

OPEN

What keeps me awake at night?



Business as unusual

How to keep up with all this?

We are not in control of technology nor technology adaption

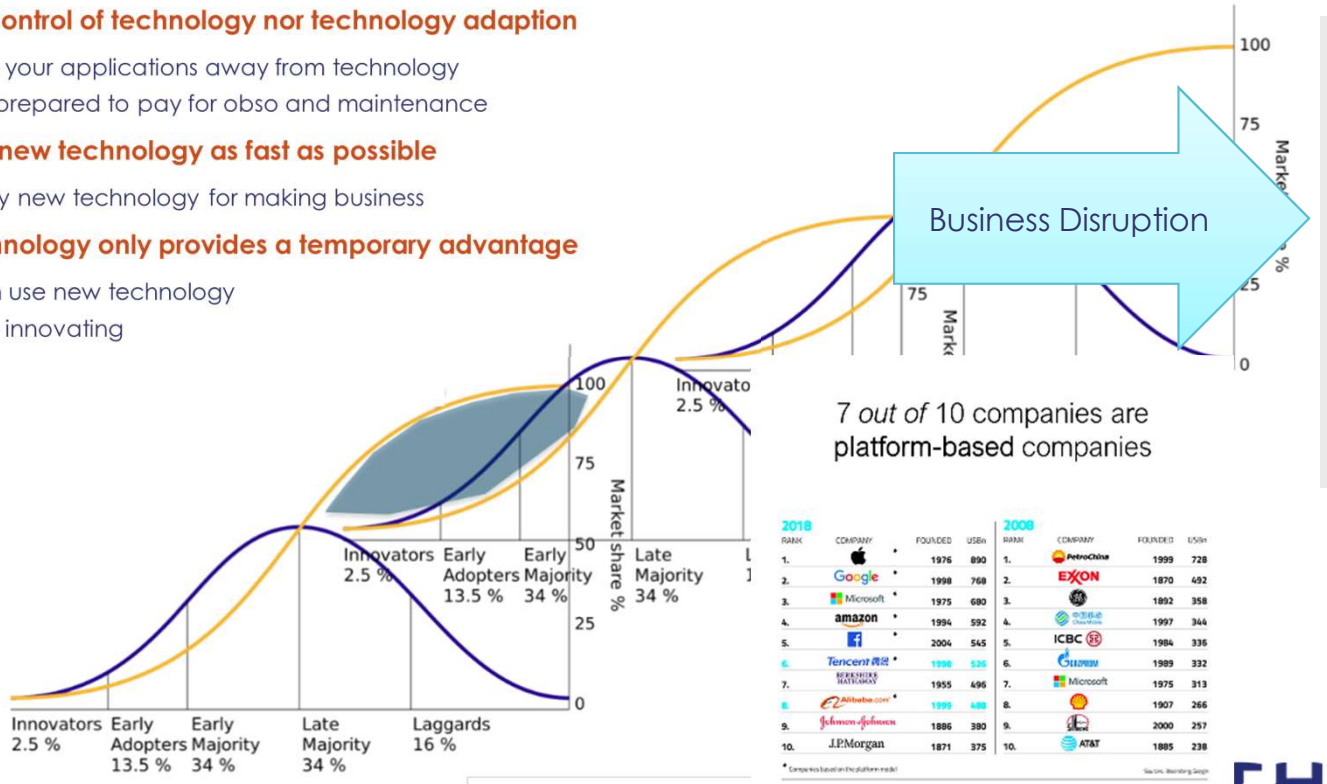
- Implying: Keep your applications away from technology
- Otherwise: be prepared to pay for obso and maintenance

Be able to use new technology as fast as possible

- Meaning: Apply new technology for making business

Using new technology only provides a temporary advantage

- Everybody can use new technology
- Implying: Keep innovating



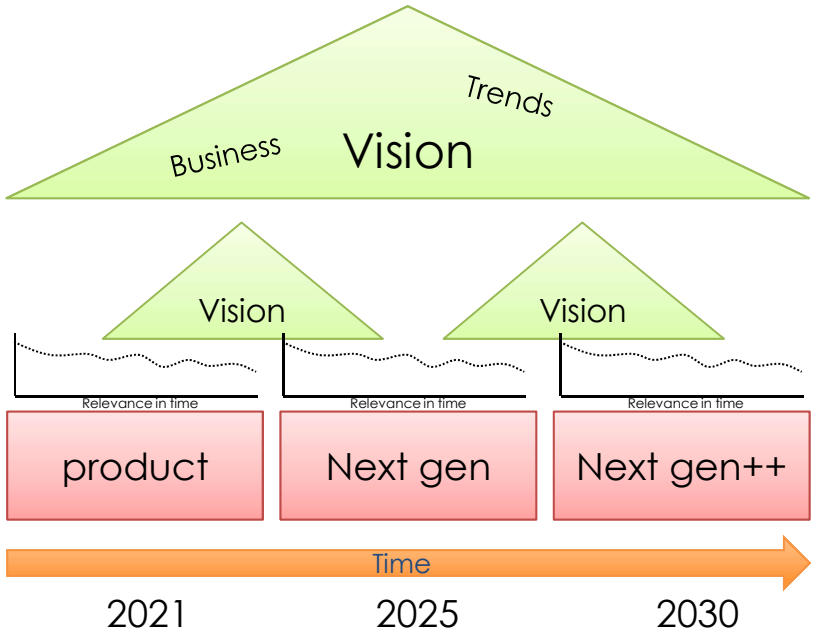
Books
Music
Films
Photography
Cars
....
Innovation
....
Radar ?
Energy?

OPEN

luminis Conversing worlds

THALES

Blending your vision and your (physical) product



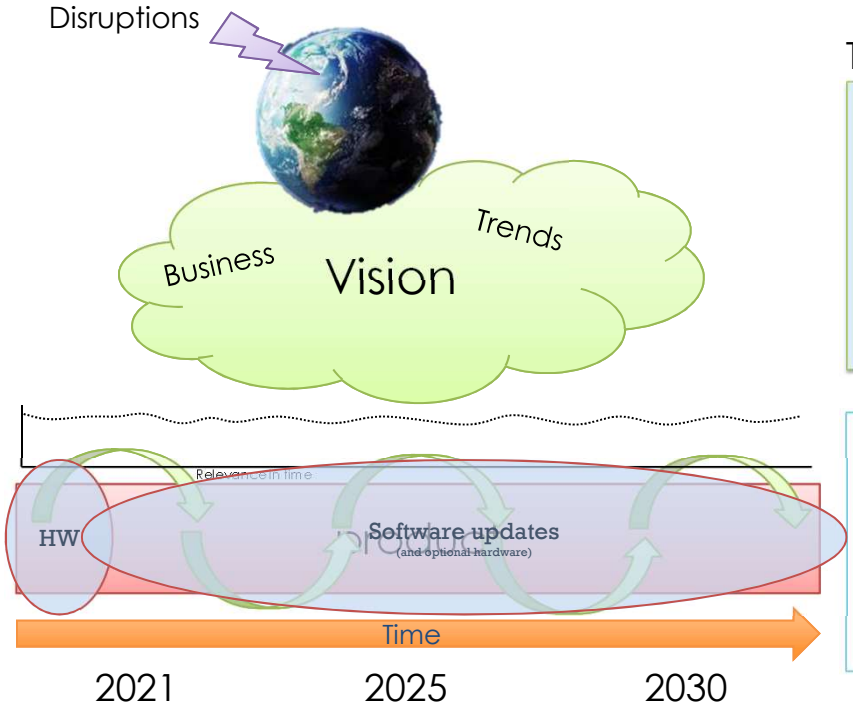
Trends (example)

Relevance
Seamless integration
Autonomous & resilient
Learning systems
Adaptive
Auto update
Cyber resilient
Coordination & orchestration
Always on
Remotely controlled
High level user interaction

Long lifecycle (30y+)
Distributed
Scaleable
Event driven
Low latency
Critical
Technology driven

THALES

Blending your vision and your (physical) product



Trends (example)

- Relevance
- Seamless integration
- Autonomous & resilient
- Learning systems
- Adaptive
- Auto update
- Cyber resilient
- Coordination & orchestration
- Always on
- Remotely controlled
- High level user interaction

- Long lifecycle (30y+)
- Distributed
- Scaleable
- Event driven
- Low latency
- Critical
- Technology dri


THE WALL STREET JOURNAL.

Home World U.S. Politics Economy Business Tech Markets Opinion Life

ESSAY

Why Software Is Eating The World

WSJ+ INVITES + OFFERS + INSPIRATION DISCOVER MORE

By Marc Andreessen

August 20, 2011

This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning its struggling PC business in favor of investing more heavily in software, where it sees better potential for growth. Meanwhile, Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised the tech world. But both moves are also in line with a trend I've observed, one that makes me optimistic about the future growth of the American and world economies, despite the recent turmoil in the stock market.

In short, software is eating the world.

Lehman's law on software evolution

> Software is subject to change in the course of its existence

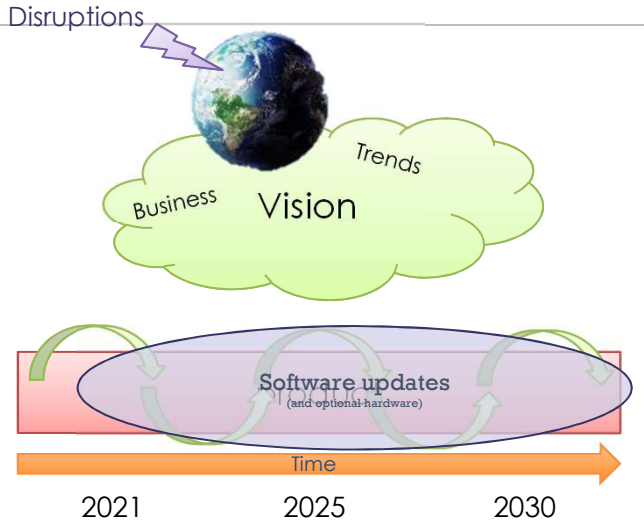
- (1974) "Continuing Change"
- (1974) "Increasing Complexity"
- (1974) "Self Regulation
- (1978) "Conservation of Organizational Stability (invariant work rate)"
- (1978) "Conservation of Familiarity"
- (1991) "Continuing Growth"
- (1996) "Declining Quality"
- (1996) "Feedback System"

source: wikipedia

OPEN

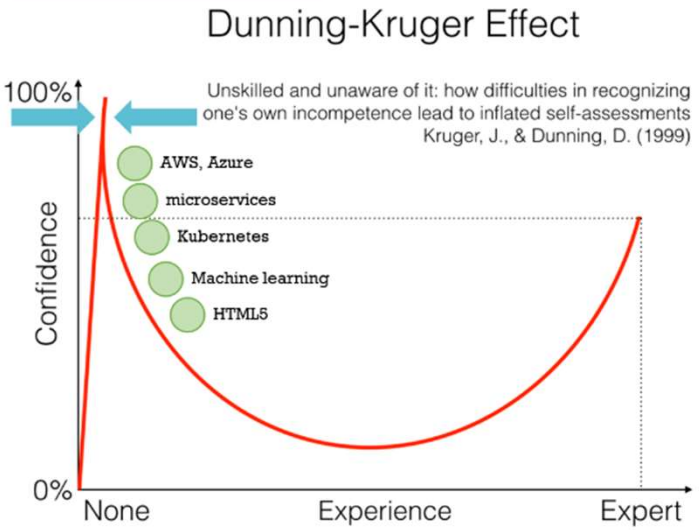
THALES

How to handle uncertainty?

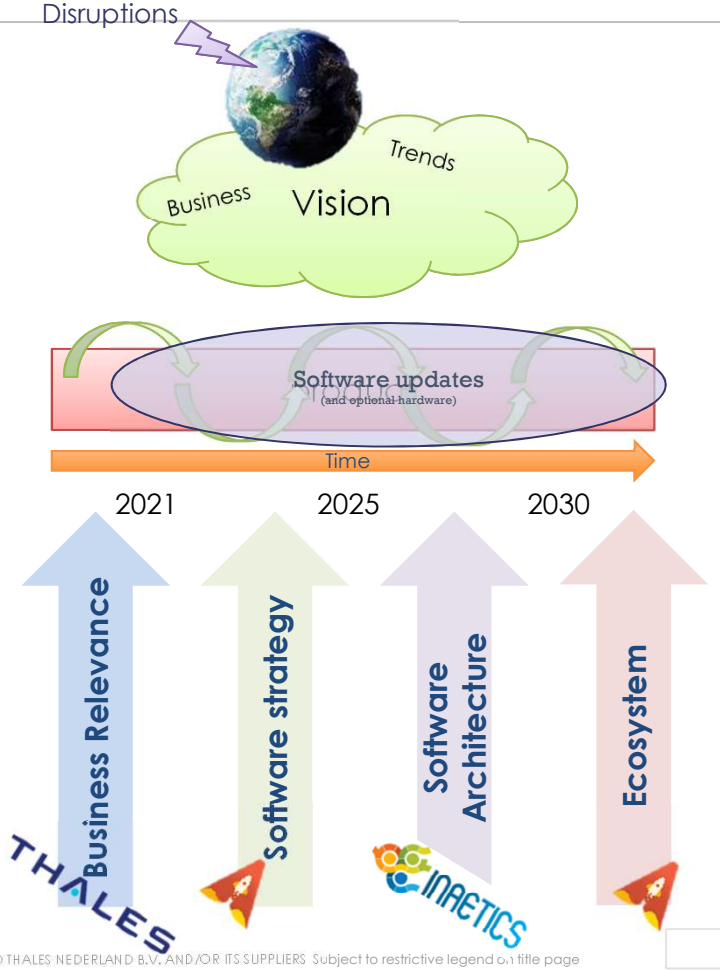


Trends (example)

Relevance
Seamless integration
Autonomous & resilient
Learning systems
Adaptive



4 pillars



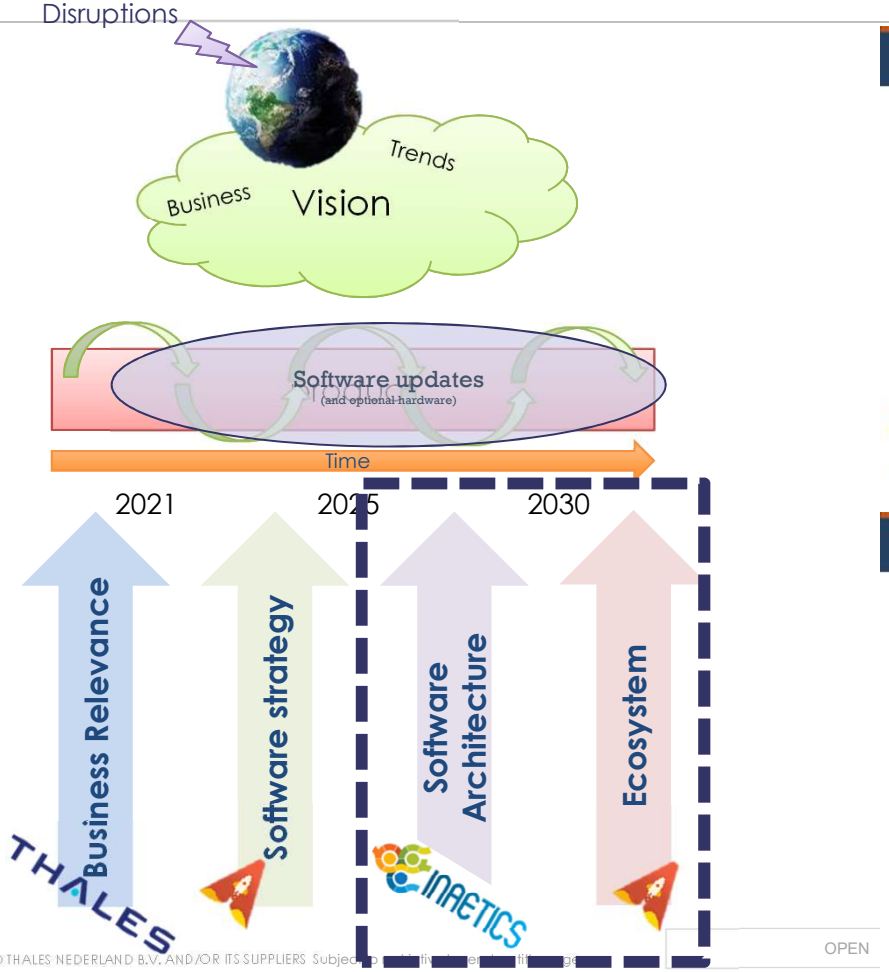
Trends (example)

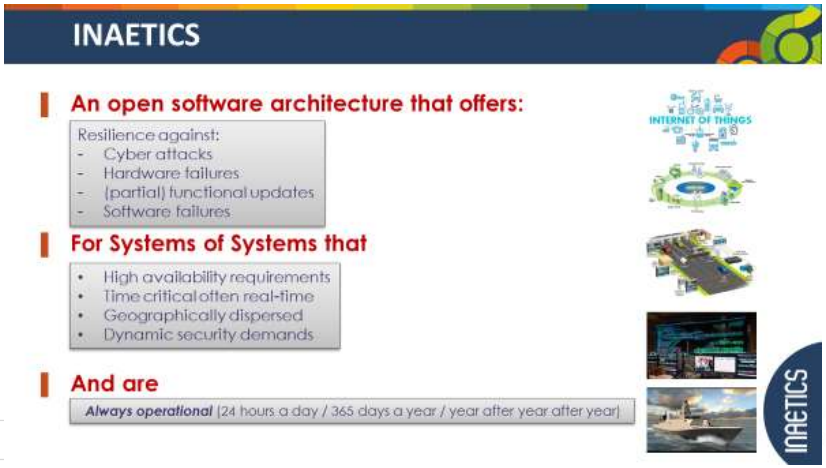
Relevance
Seamless integration
Autonomous & resilient
Learning systems
Adaptive
Auto update
Cyber resilient
Coordination & orchestration
Always on
Remotely controlled
High level user interaction

Long lifecycle (30y+)
Distributed
Scaleable
Event driven
Low latency
Critical
Technology driven

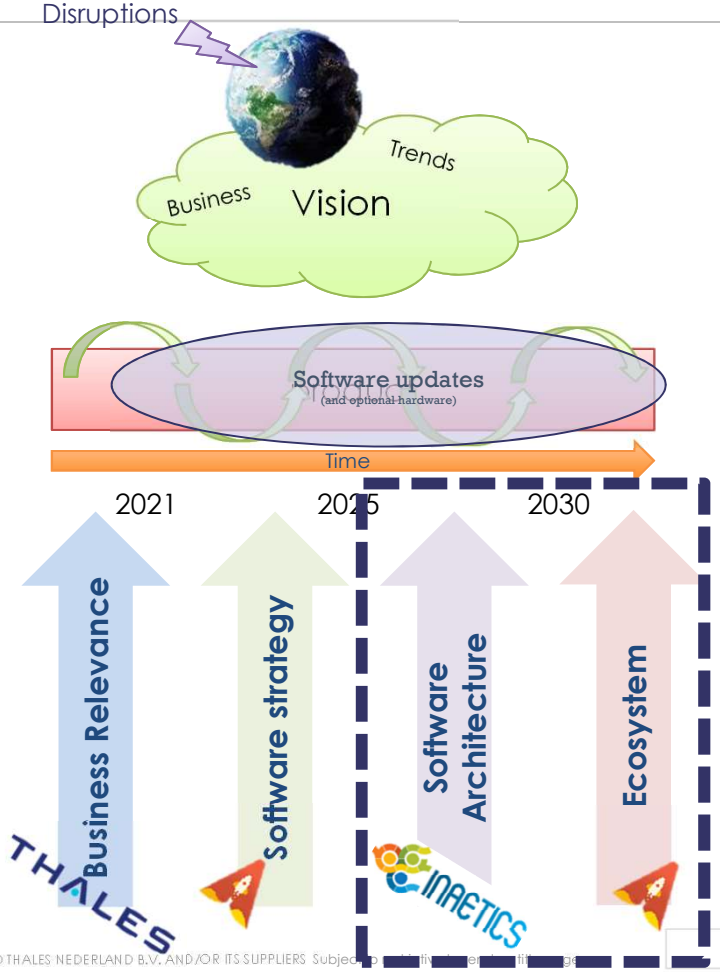
THALES

4 pillars: Software Architecture (INAETICS #1)





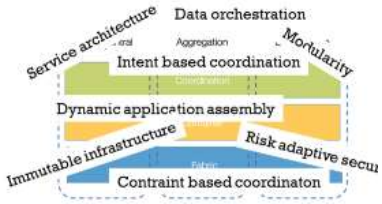
4 pillars: Software Architecture (INAETICS #2 2023 (?))



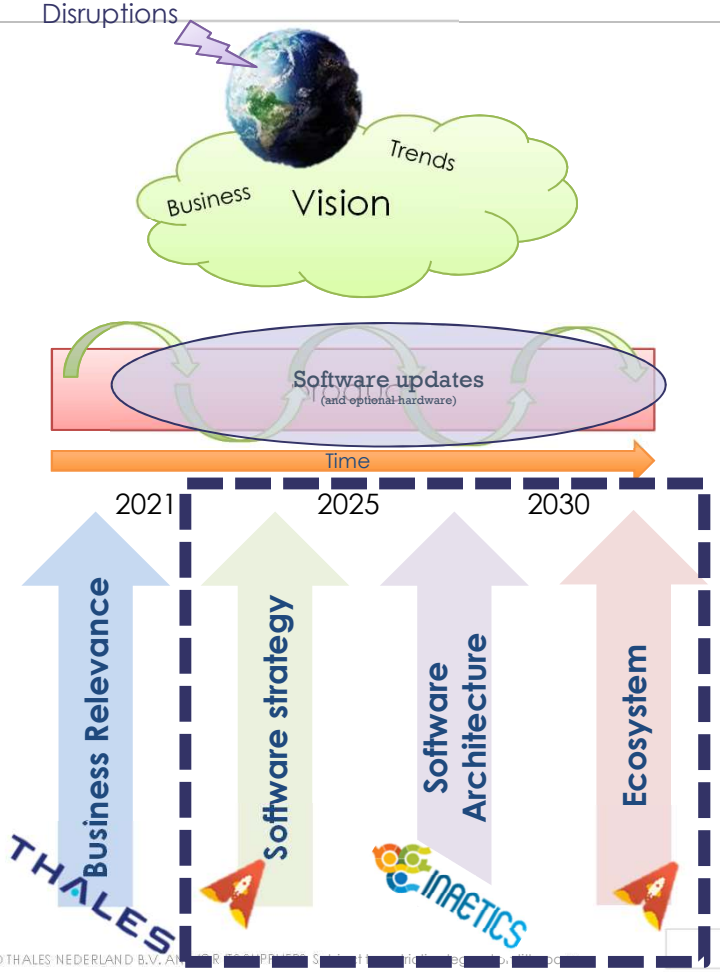
Inconvenient architectures...

Modern characteristics:

- Systems are switched on once
- Security is risk-aware
- Performance is elastic
- Systems are resilient and evolvable



4 Pillars: Technical Leaders (Accelerate)



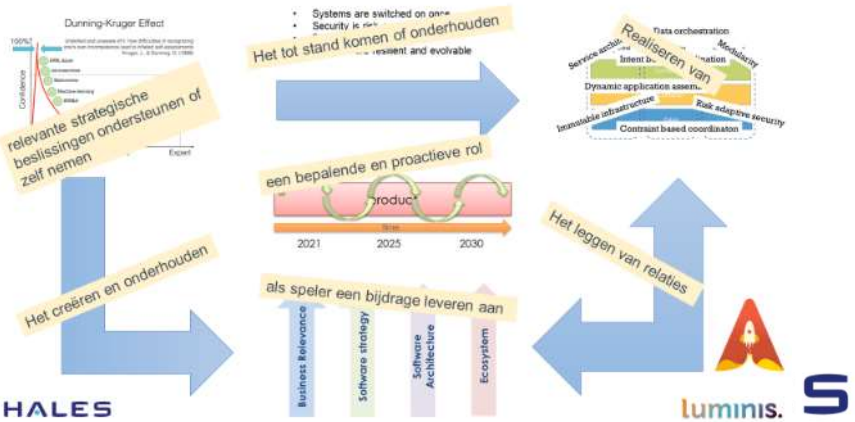
What is accelerate?

In accelerate...

- Education program for technical leaders in the software domain
- We comprimized 5 years of professional development in a program of 18 months
- Top talents make the shift to future (technical) leader

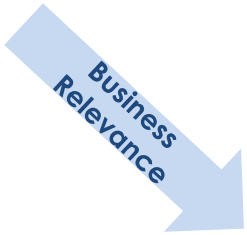
How?

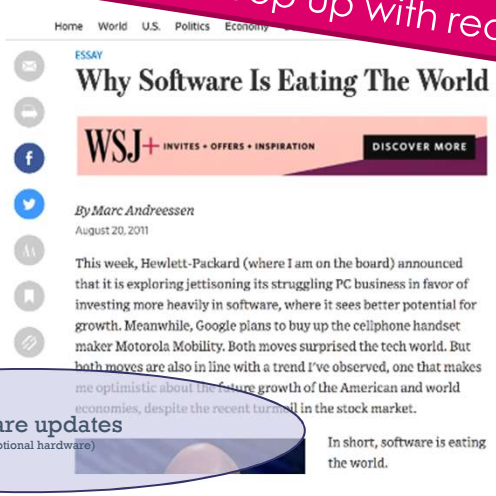
- Improving soft-skils with training and coaching on the job
- Broadening and deepening technical knowledge and skills
- Becoming an architecture expert
- Put the theory into practice!
- Extend their network



Thank you for your attention

How to keep up with reality?





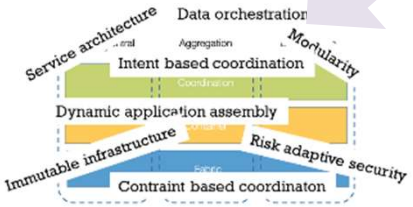
Long lifecycle (30y+)
Distributed
Scaleable
Event driven
Low latency
Critical
Technology driven

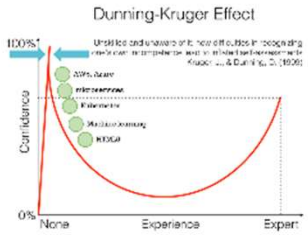
Ecosystem

Software updates
(and optional hardware)

Software strategy

Software Architecture





Lehman's law on software evolution

> Software is subject to change in the course of its existence

- (1974) "Continuing Change"
- (1974) "Increasing Complexity"
- (1974) "Self Regulation
- (1978) "Conservation of Organizational Stability (invariant work rate)"
- (1978) "Conservation of Familiarity"
- (1991) "Continuing Growth"
- (1996) "Declining Quality"
- (1996) "Feedback System"

source: wikipedia

luminis
Conversing worlds

THALES

OPEN