inventors for the digital world

AMDT (Action Mutualisée de Développement Technologique)

T. Kloczko, D. Rey

Introduction

The Inria SED are pioneers since 2003

No similar organization / service in other academic institutes

No similar aims (scientific software) in industry

Main experimentations from 2003 to today (Crisam)

Short ADT (3-6 months)

IA / IJD / IC supervision by SED engineers

Longer ADT (12-24 months)

ADT with 2 engineers / 2 ADT for 1 engineer

ADT PFE (software and/or harwdare)

Agile methodology (Scrum, extrem programming)

AMDT

4 pillars seem to be essential for software development in our context

Project mode

Code and know-how factorization

Agility

Team

01

High level software

Inria missions

« Scientific excellence…

Computer science & applied mathematics

… for technology transfer and society »

Impact is important

Partners expectations

Partners:

Academic or industrial partners

Computer science, but more often other domains

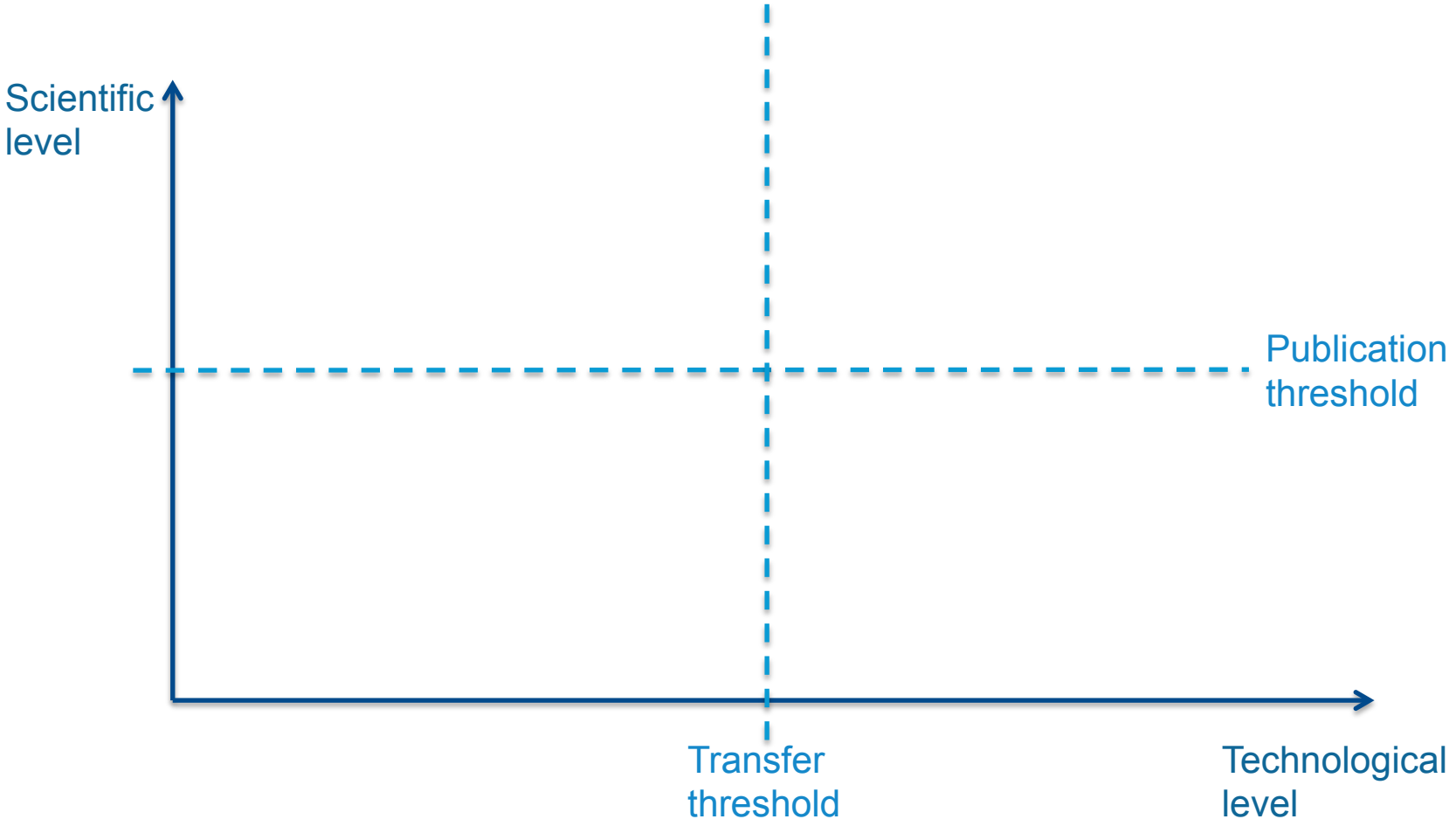
- Biology, medicine, energy, …

Expectations:

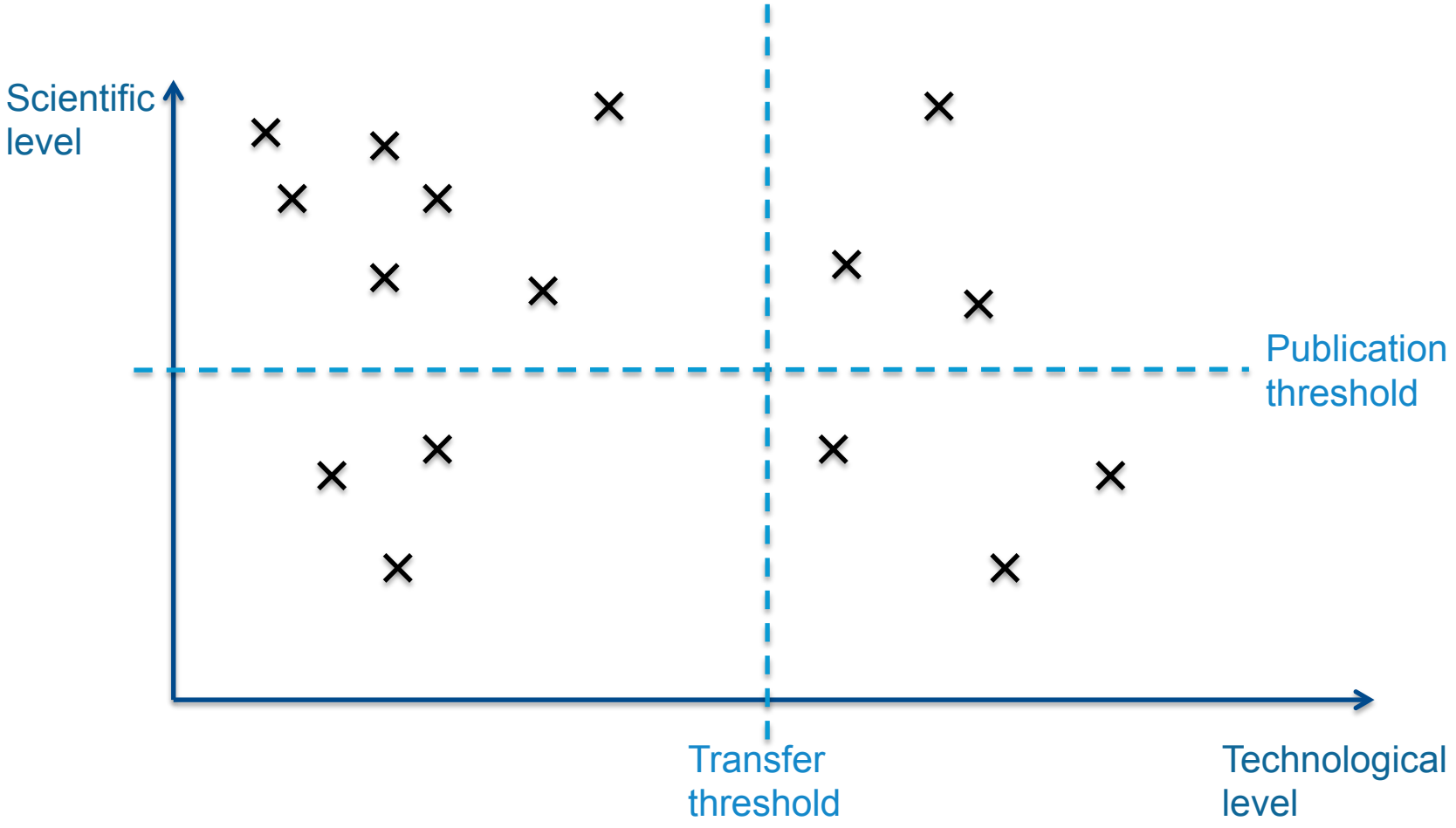
Know-how, Co-supervision, Co-publication

High-level software (patent sometimes)

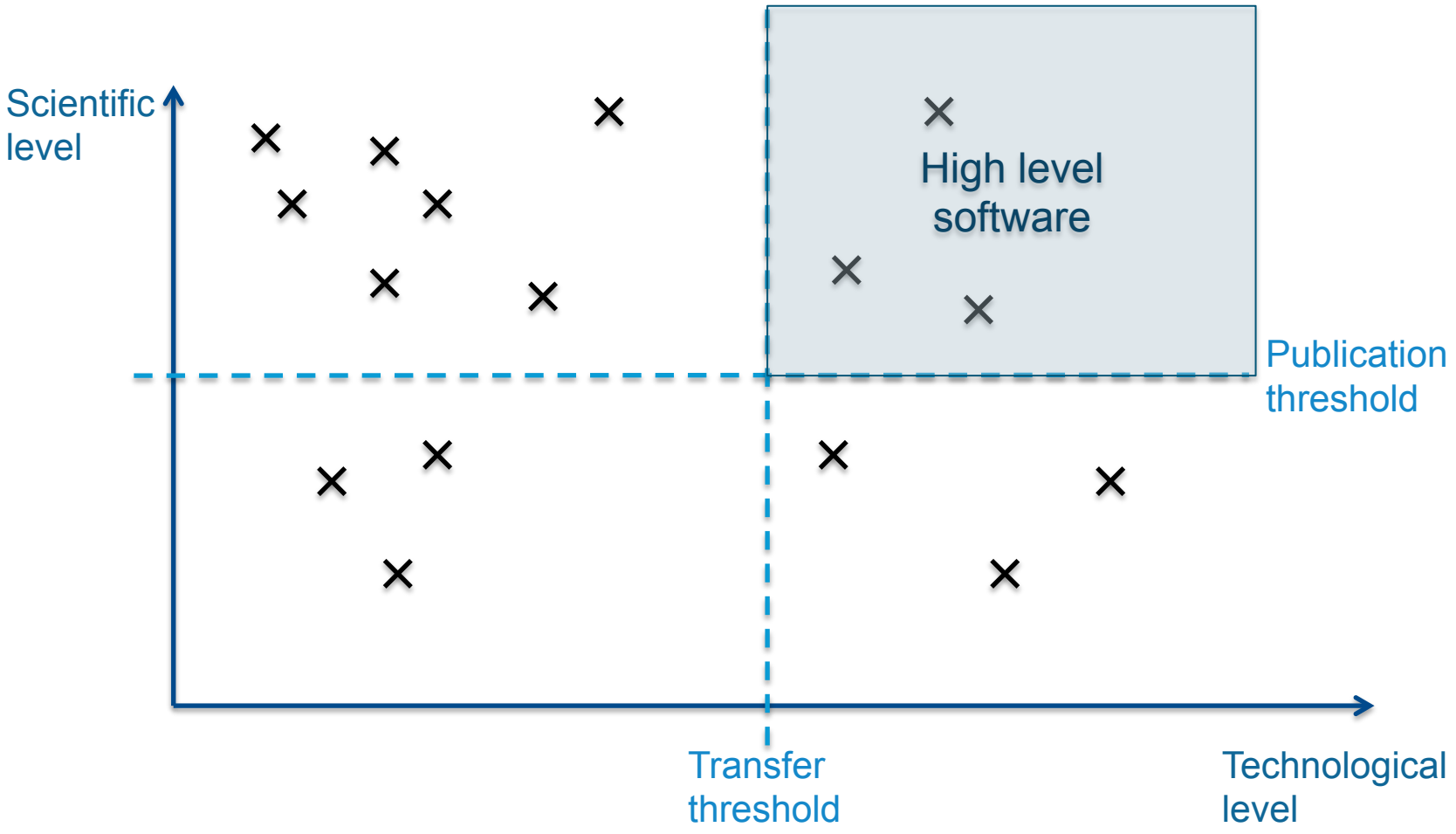
High level software



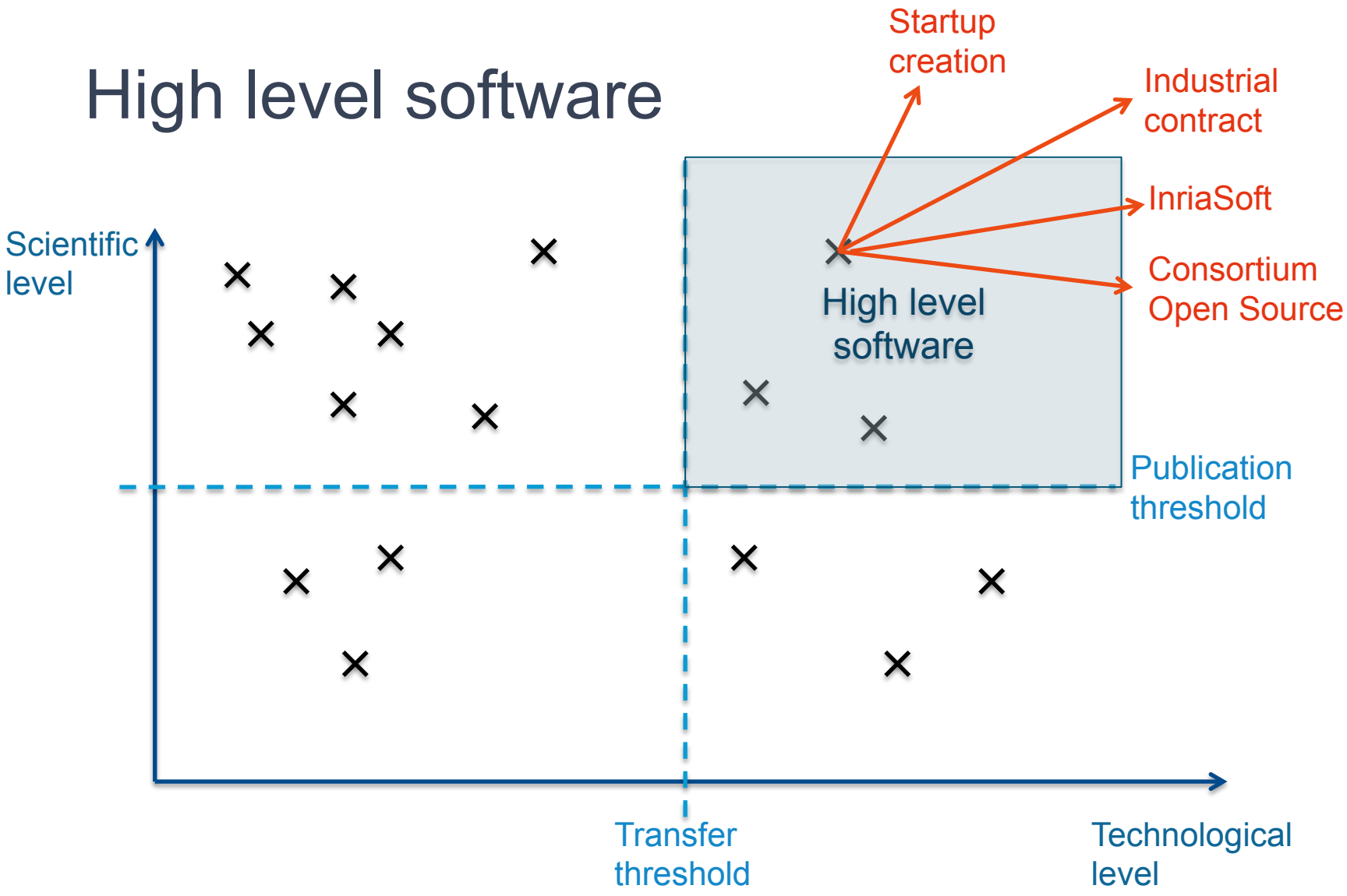
High level software



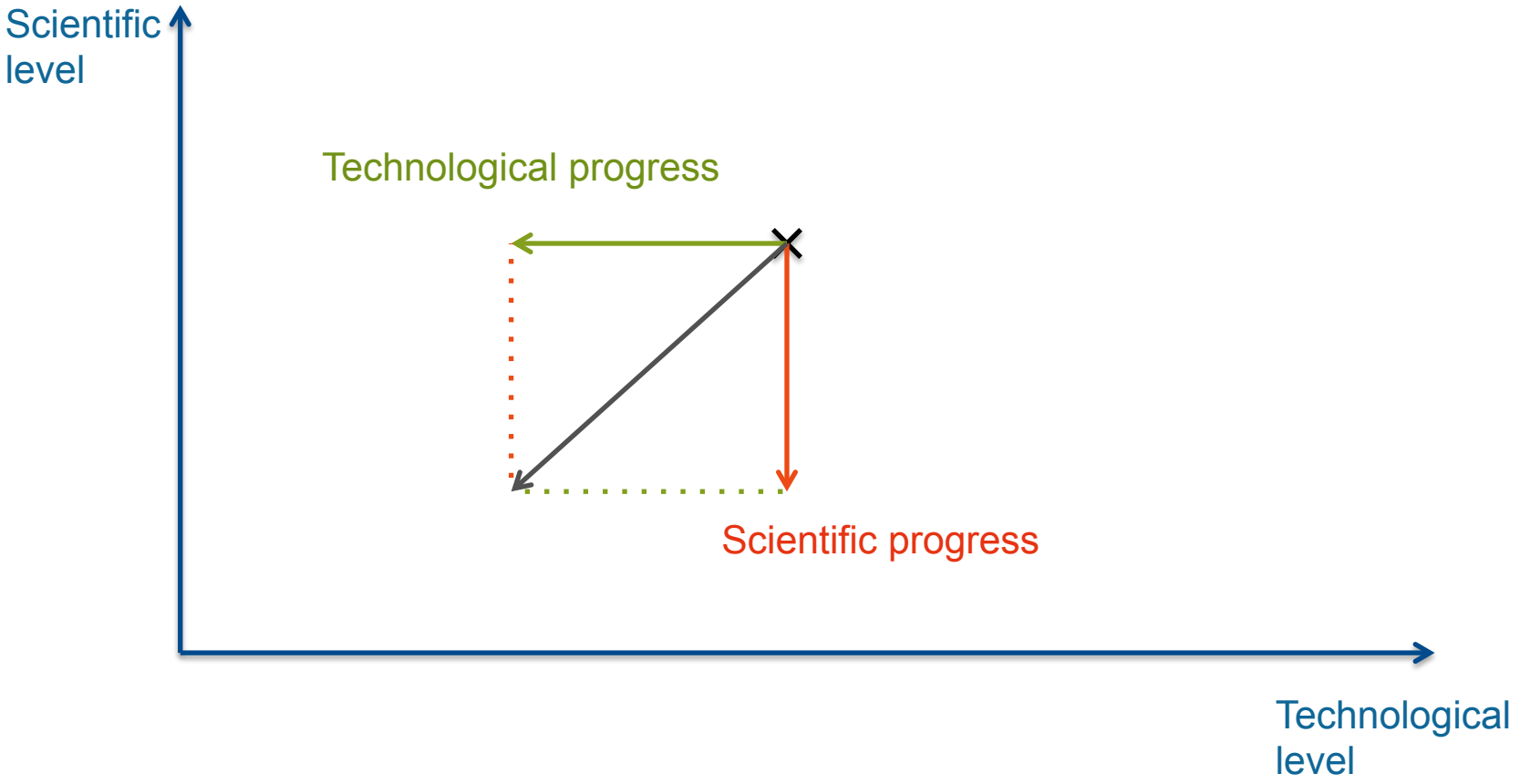
High level software



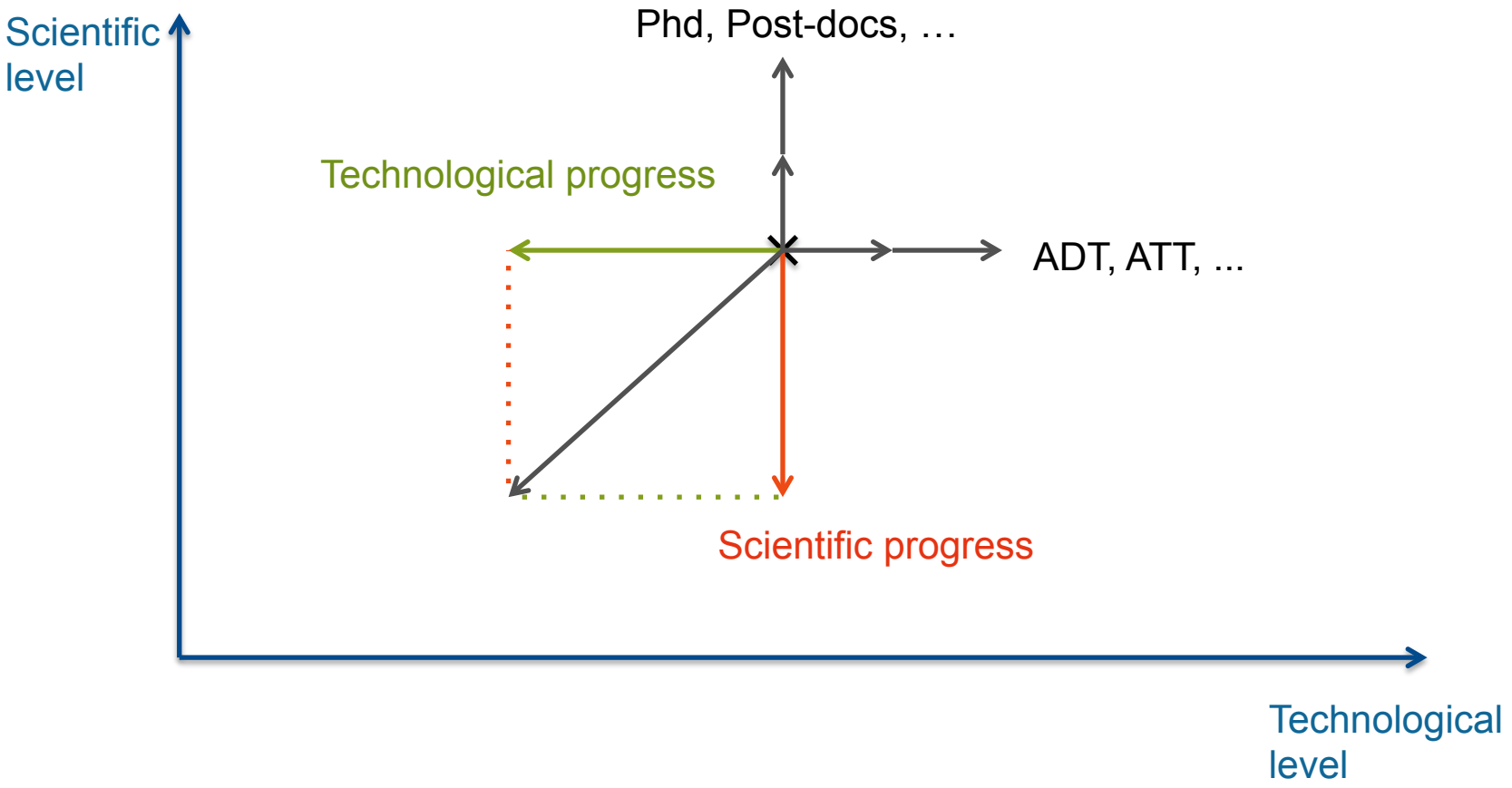
High level software



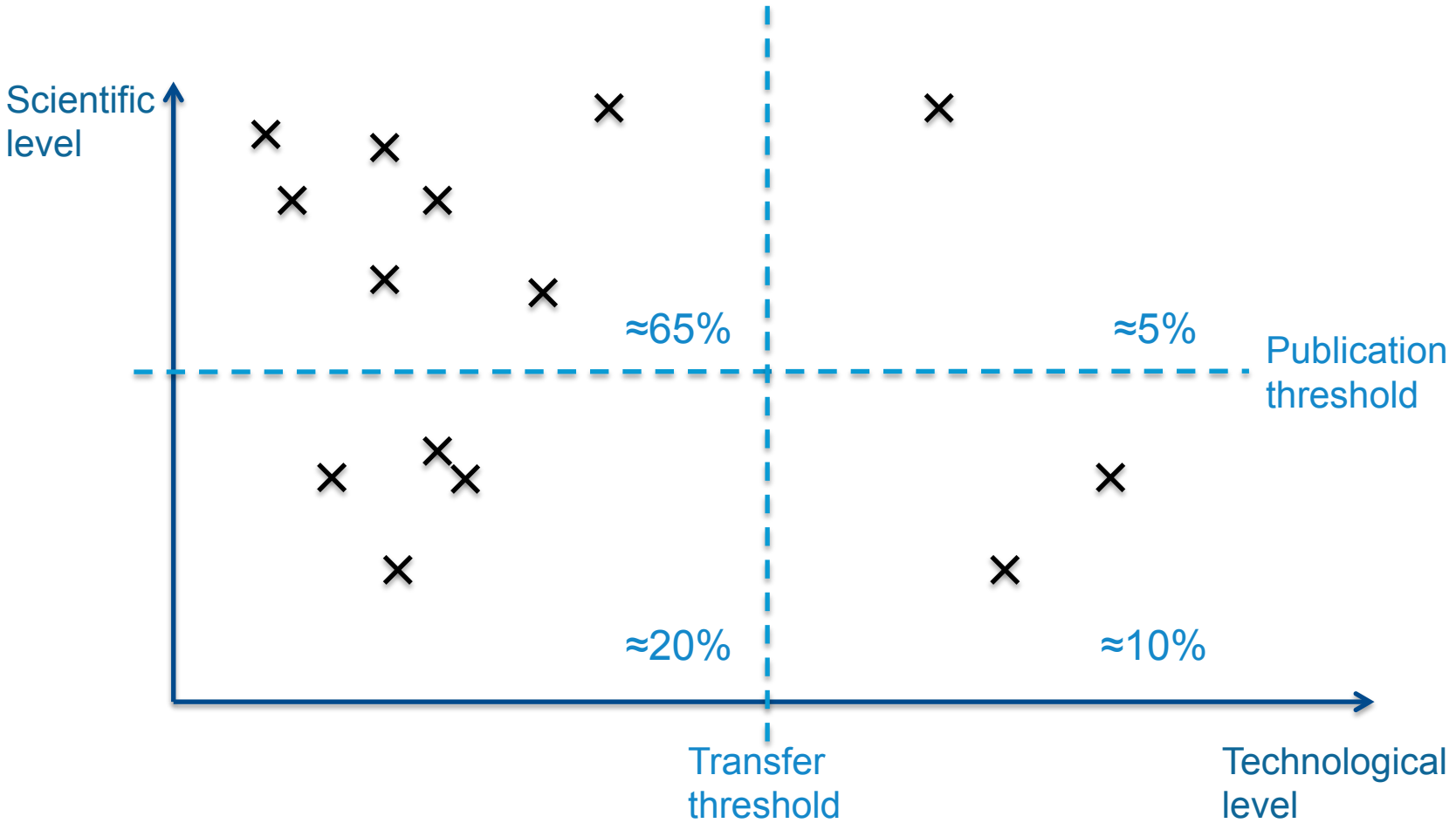
Temporal obsolescence



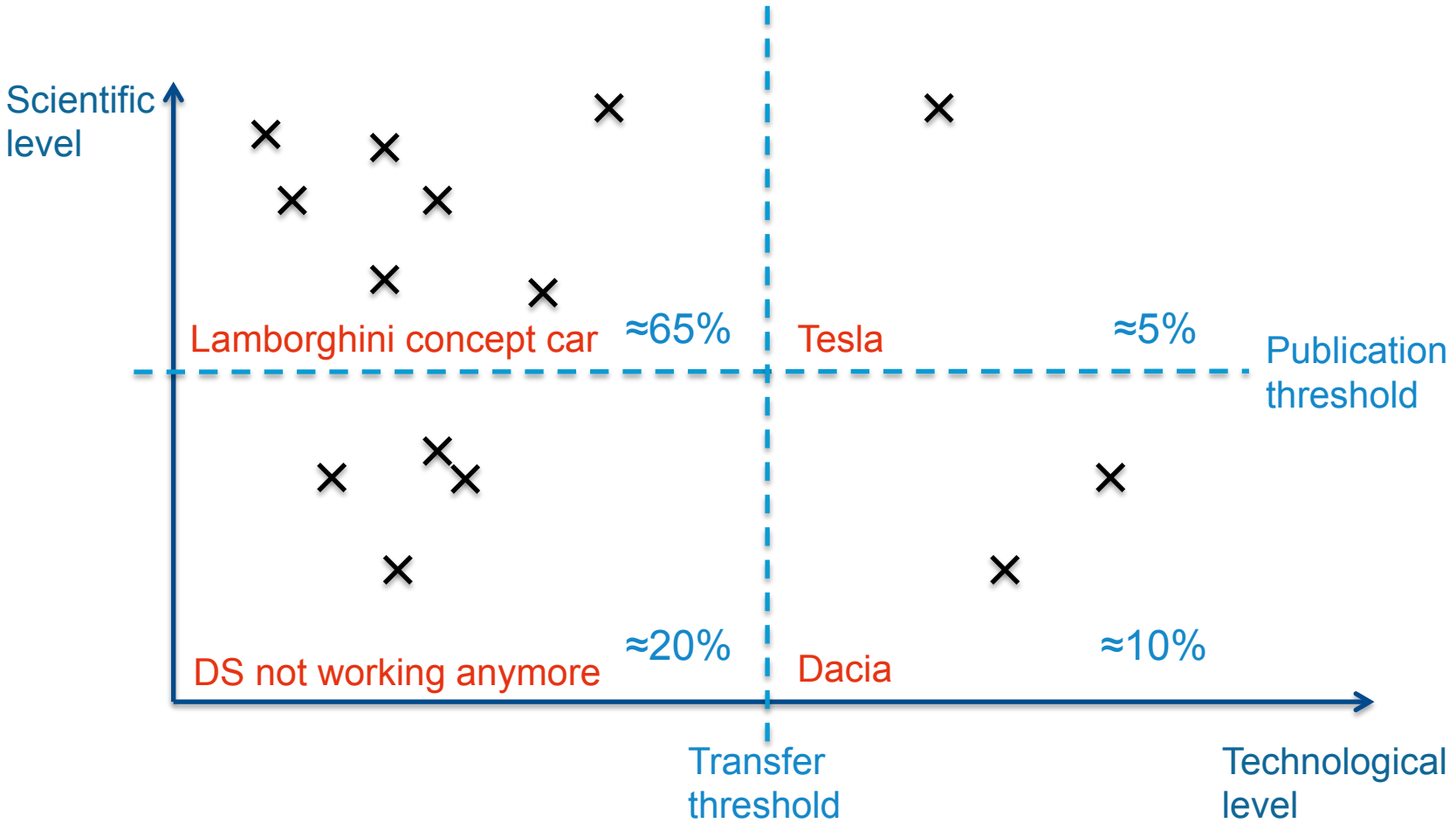
Temporal obsolescence



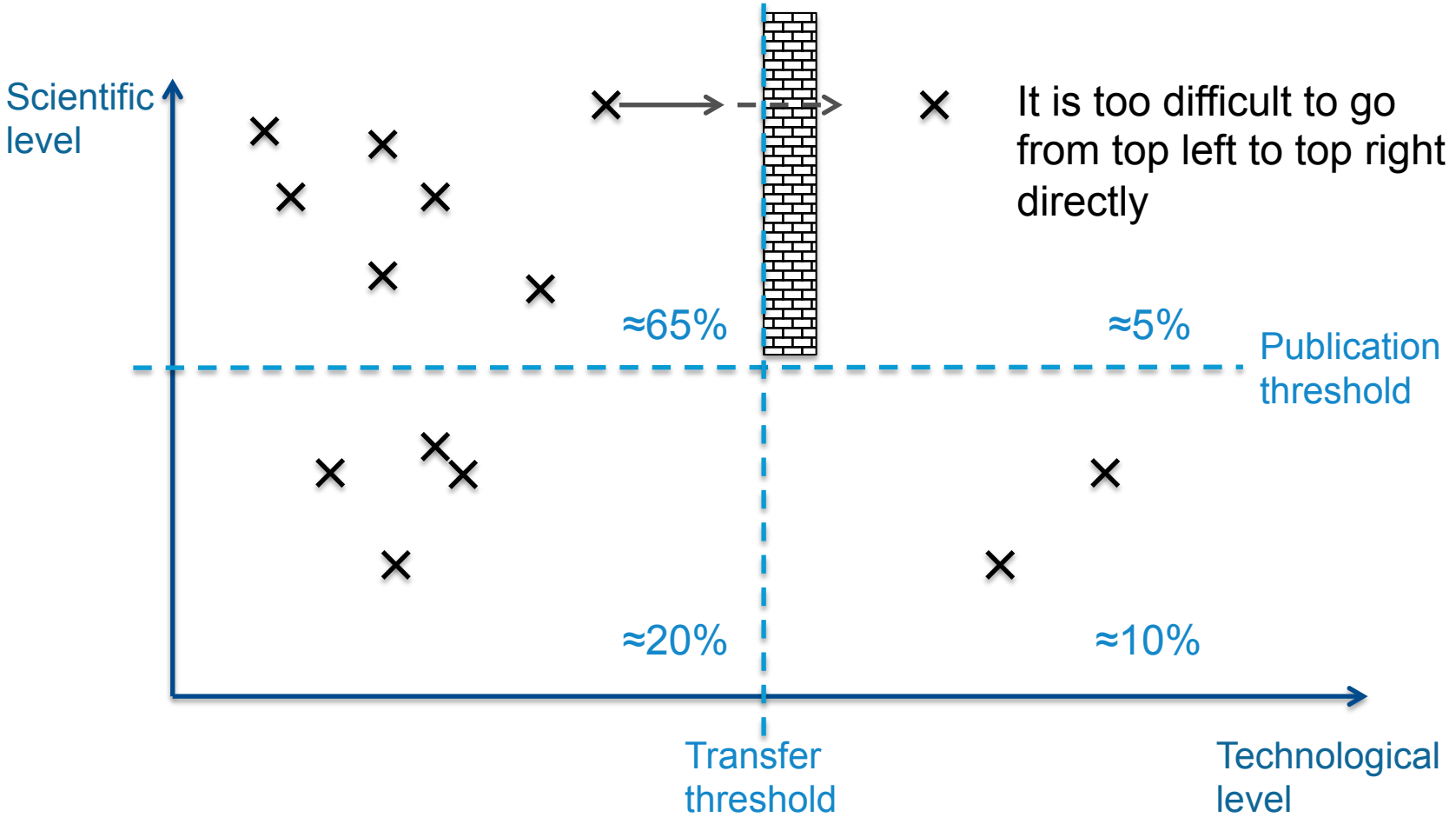
Observations at Inria



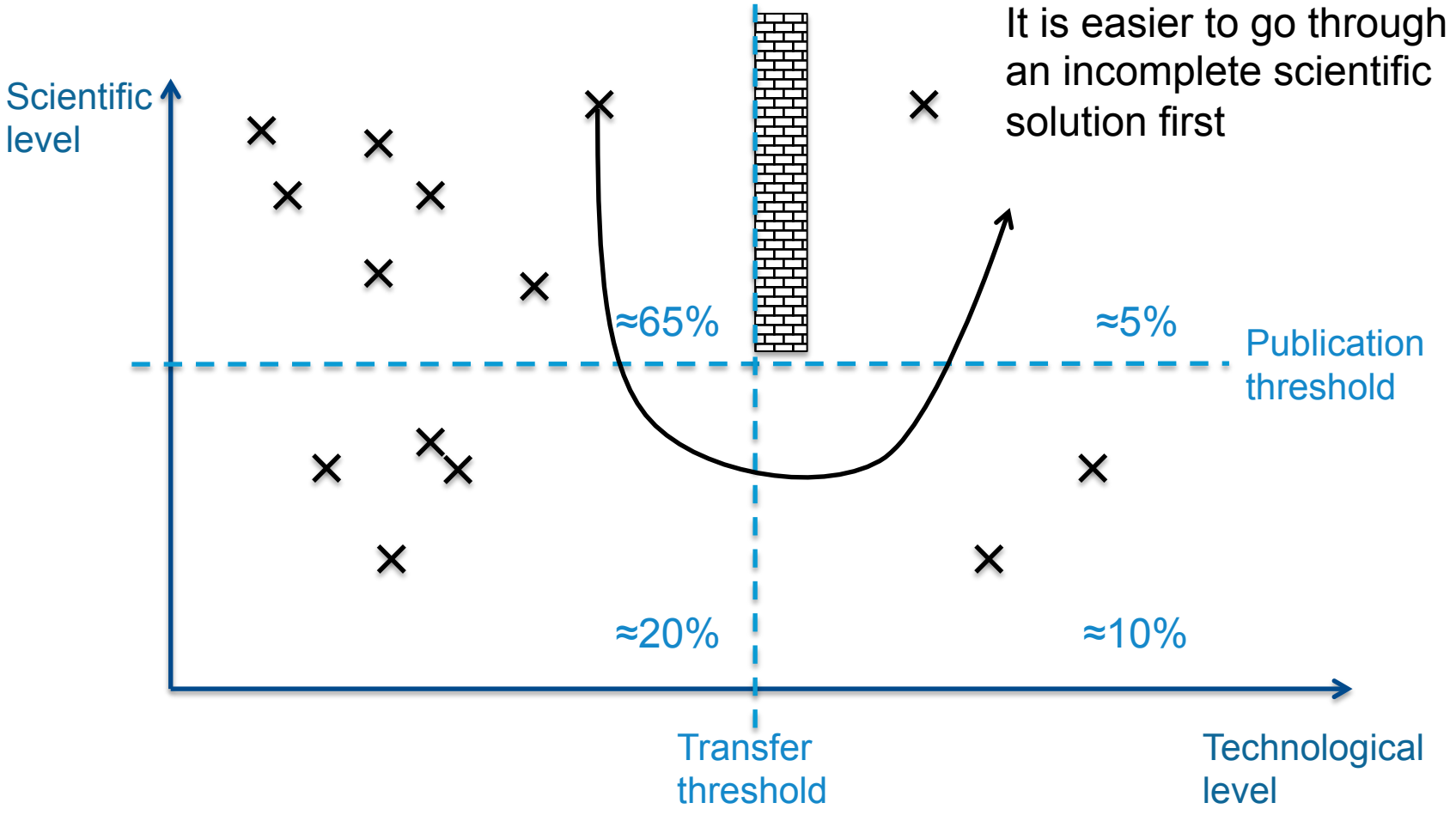
Observations at Inria



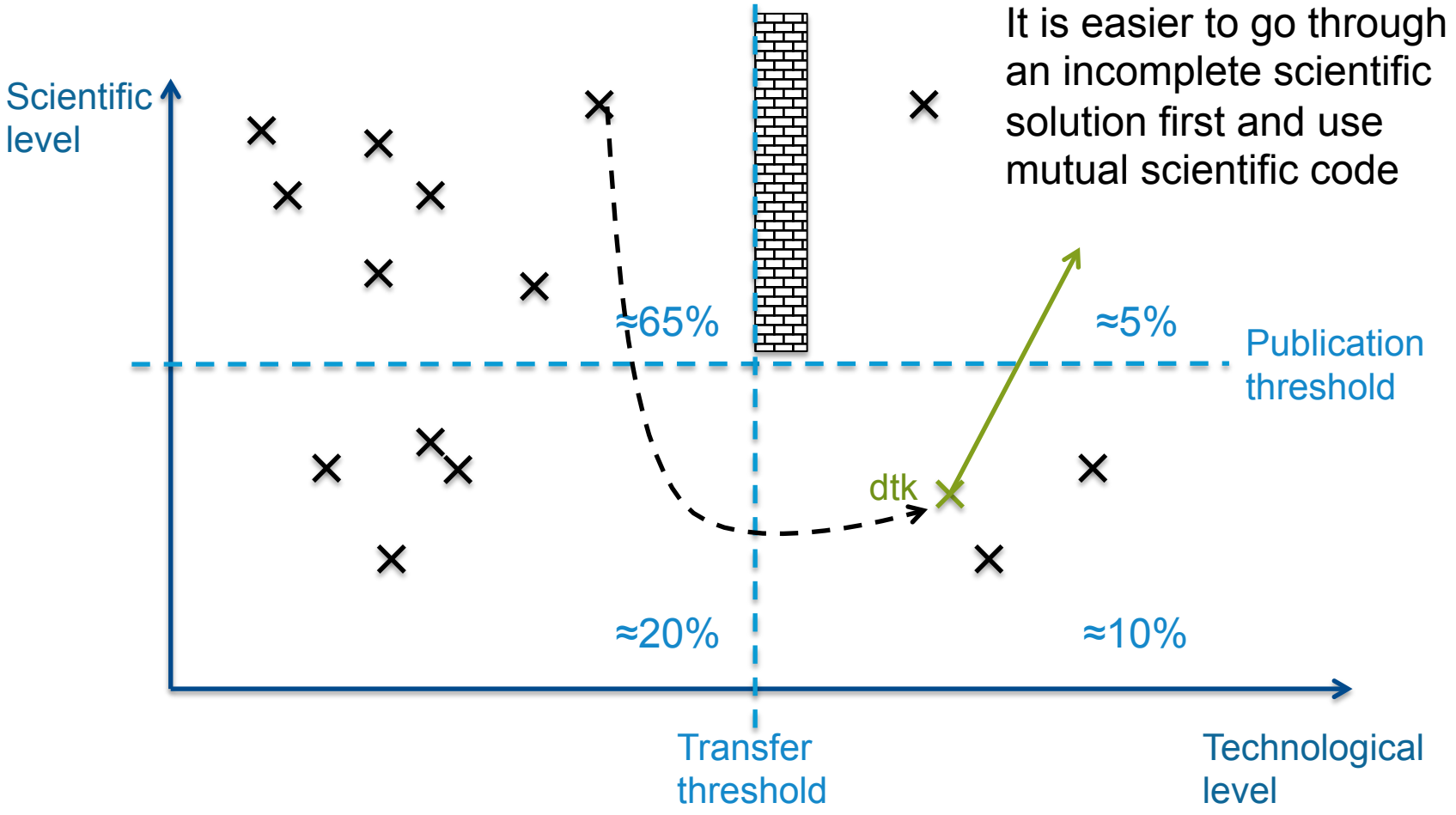
Observations at Inria



Observations at Inria

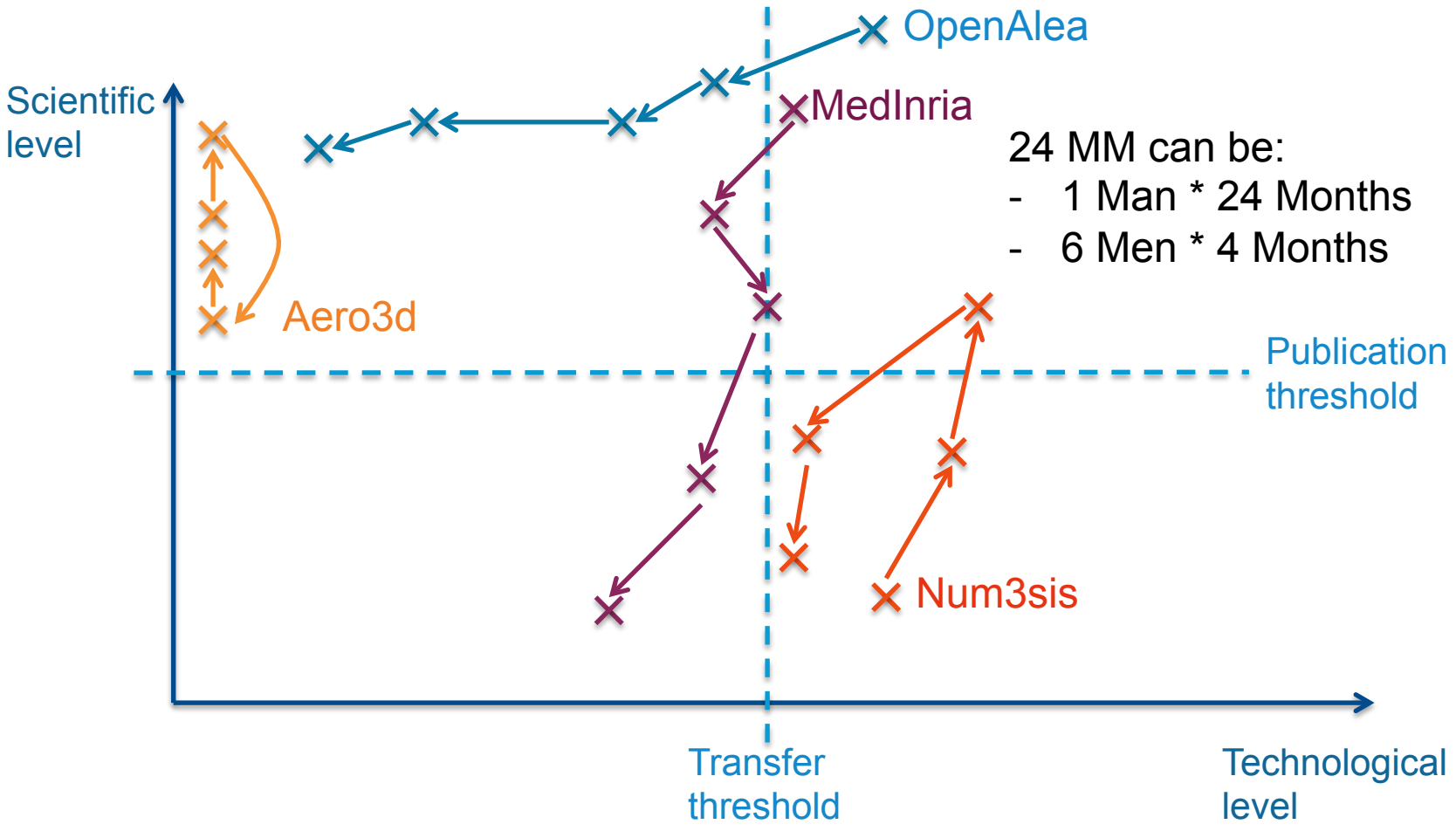


Observations at Inria



It is easier to go through an incomplete scientific solution first and use mutual scientific code

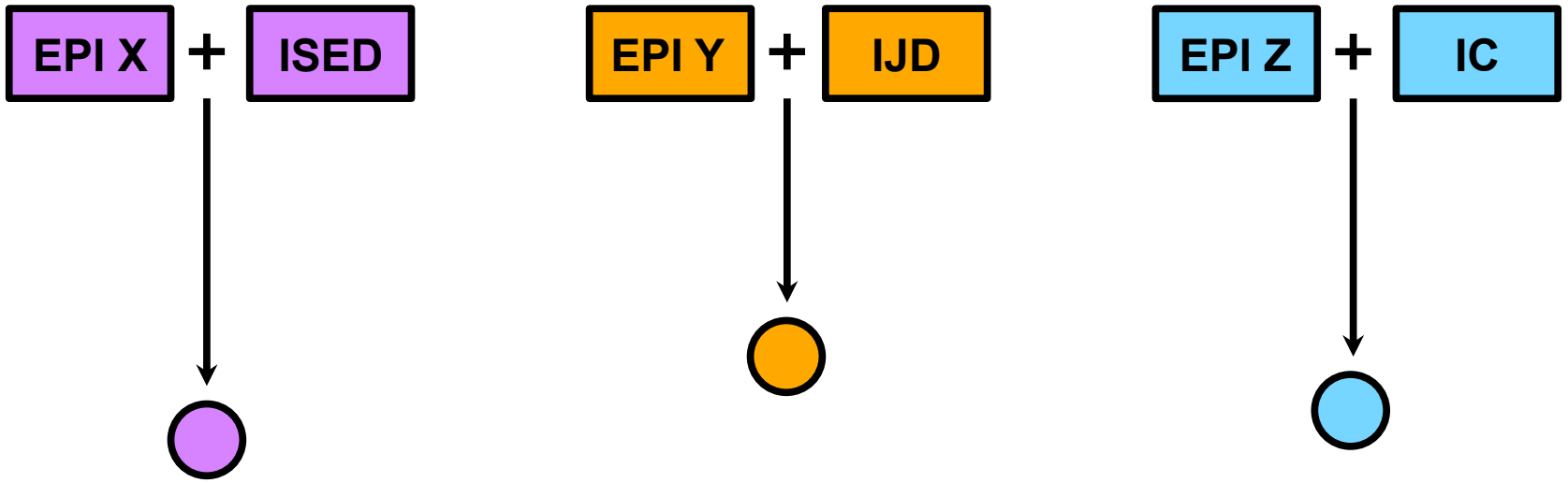
Observations at Inria: exemples



02

AMDT and dtk

Before 2012: separated ADT



Vertical development (work in silo)

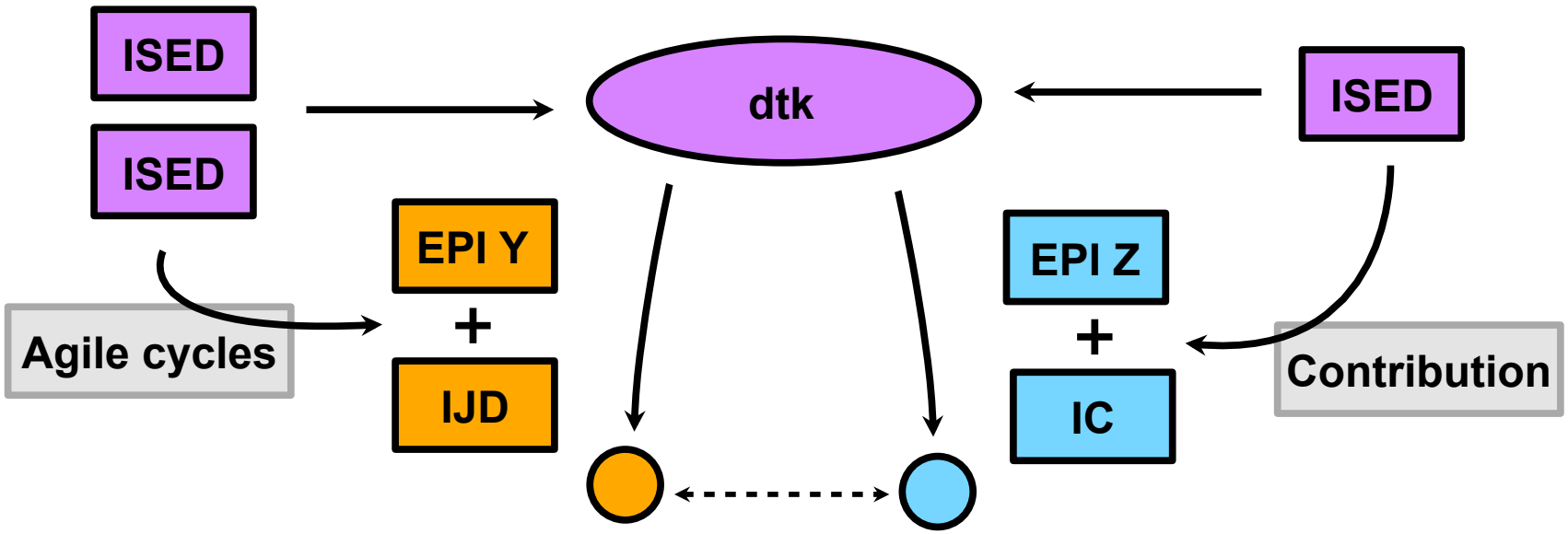
Unequal quality, poor following of CDD engineers by the SED

High dependency to recruited engineer for each ADT

Poor interoperability between ADT results

Long term maintenance not possible

From 2012 to 2016: PFE dtk



Vertical & horizontal development

Quality of the codes is increased and code is factorized

Less dependency to the CDD engineer

Minimal interoperability

Minimal long term maintenance is possible

Since 2017: AMDT

Agility

Mix between Scrum and Extrem Programming

Timeline is baseline

Incremental functional software

Clients (researchers) are involved a lot

Team

4-9 engineers (4-5 SED 2-4 CDD): multidisciplinary and complementary profiles

1 scrum master for each cycle

Developpers in the EP can join the team

Everyone works on every projects

AMDT and InriaHub

ADT to add CDD engineers on the team

ADT to decide how long SED engineers are involved

Since 2017: AMDT

Agility

Mix between Scrum and Extrem Programming

Timeline is baseline

Incremental functional software

Clients (researchers) are involved a lot

Team

4-9 engineers (4-5 SED 2-4 CDD): multidisciplinary and

1 scrum master for each cycle

Developpers in the EP can join the team

Everyone works on every projects

AMDT and InriaHub

ADT to add CDD engineers on the team

ADT to decide how long SED engineers are involved

InriaHub 04/2016

TissueLab (VirtualPlants)	– 12 HM
WindPos (Tosca)	– 12 HM
Patient Monitoring (Stars)	– 18 HM
FlowNext (Acumes)	– 08 HM
MGDA Export (Acumes)	– 02 HM
BCI Browser (Athena)	– 12 HM
MultiDomain platform (SED)	– 18 HM

InriaHub 07/2016

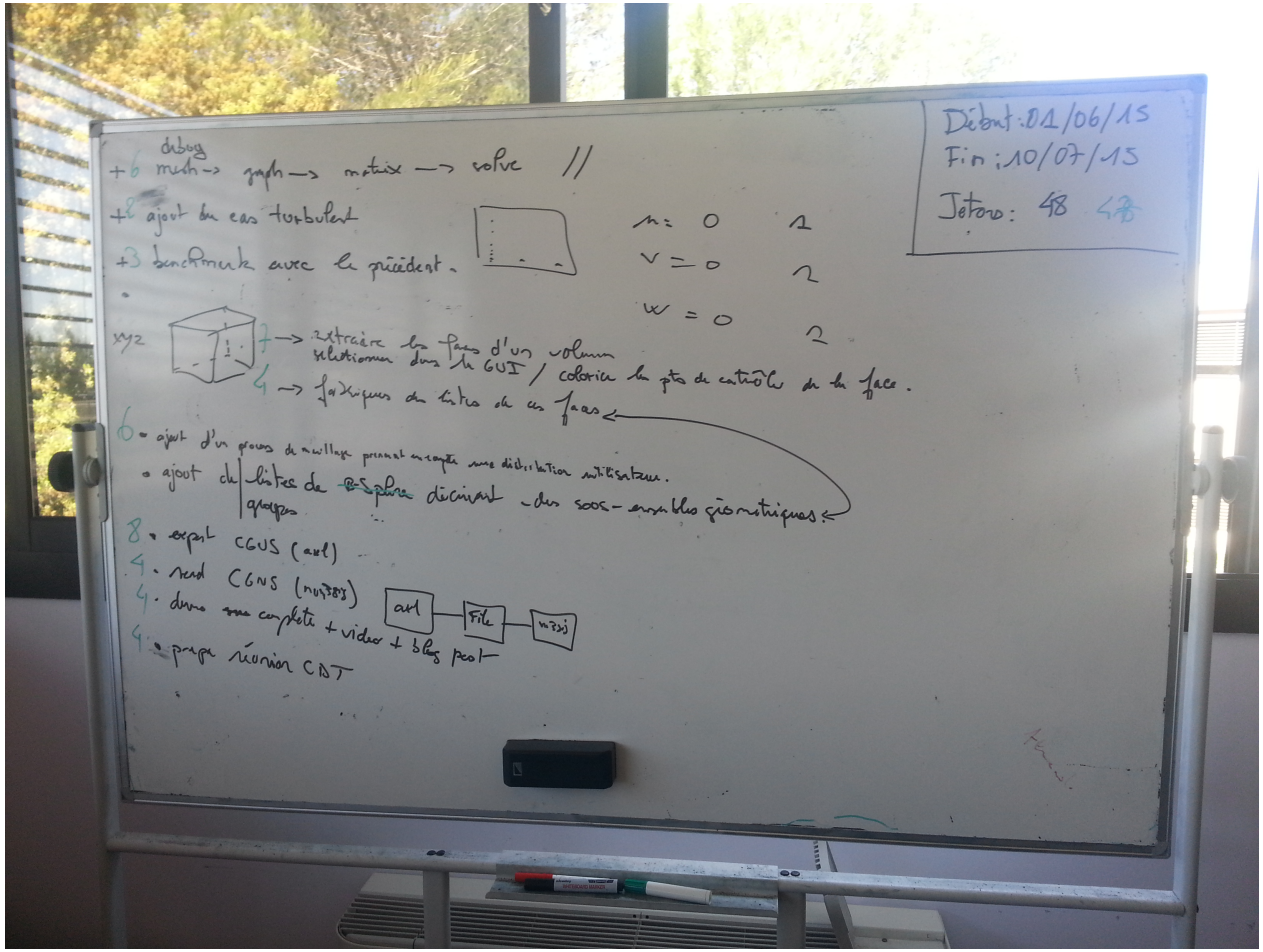
Odin+ (Biocore)	– 24 HM

InriaHub 05/2017

SW-Plateform (Lemon)	- 06 HM
Bolis2 (Apics)	- 06 HM

+ CDT following over time

AMDT in practice



AMDT in practice

AMDT + dtk main advantages 1/2

Human resources

Specialized and complementary engineers

Less problems with individual lack of competencies or anticipated contract end

Possible reallocation of resources if changes in a project planification

Team is globally becoming better over time thanks to domain coverage

Team work allows to increase the scientific and technical levels of a software

Where individuals generally don't (technical and scientific progresses are too strong)

Temporal cycles

Incremental functional solutions (software is always working)

Planning for the next 2 months is fixed (cycle change is known in advance)

EP implication is eased

EP has to be active in each cycle

At least: cycle beginning (features priority) and demo meeting at cycle end

EP developers can easily complete the team in a given cycle

AMDT + dtk main advantages 2/2

Software production

Production time and cost reduction

Applications quality increase

Up-to-date technologies

Capitalization of code and know-how in the SED and in dtk

Diffusion and transfer impact

Community building is ease

Many functional demonstrators are produced to show Inria know-how

More professional production processes and results

Productivity is increased naturally

Example without AMDT: for each ADT, at least 3 months of initial learning curve…

Example without AMDT: each ADT several months of useless development (debug time, useless feature, …), at least 3 months of waste of time

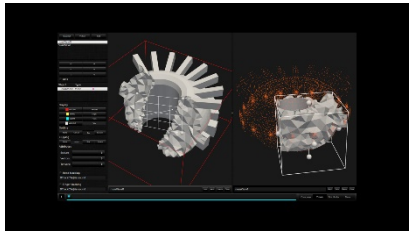
03

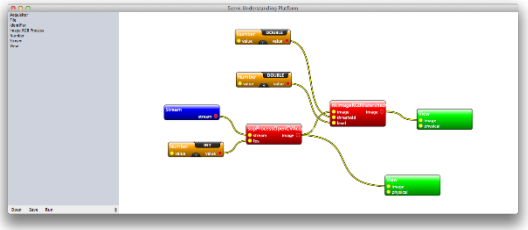
dtk

dtk overview

Key figures

dtk Kernel	9 generic layers (> 50K lines of code C++)
dtk Thematic Layers	9 layers dedicated to specific domains (imaging, geometry, linear algebra) (> 40KLoc C++)
dtk Thematic Plugins	7 groups of plugins implementing thematic layers (> 47KLoc C++)
dtk-based Platforms	11 teams' platforms (gnomon, fs3d++, sup, num3sis, axel, medInria, pib, enas, windpos, inalg@e)





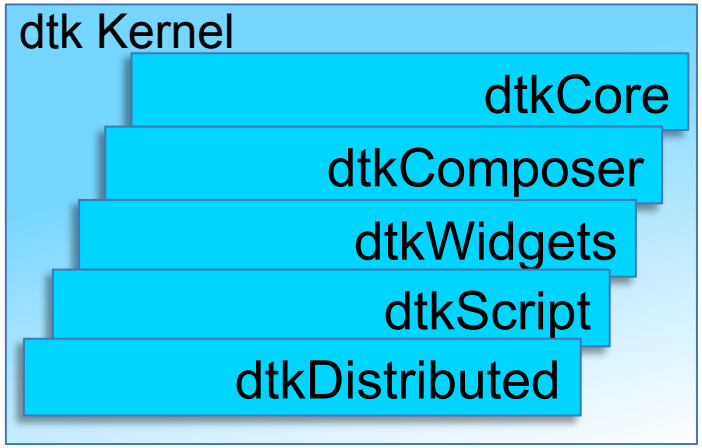
Organization

team: 3 permanent research engineers (2012 - 2016 via ADT plateforme), AMDT team since 2017

dtk-committee: 5 permanent researchers involved in teams' platforms and related to the local CDT

dtk Kernel

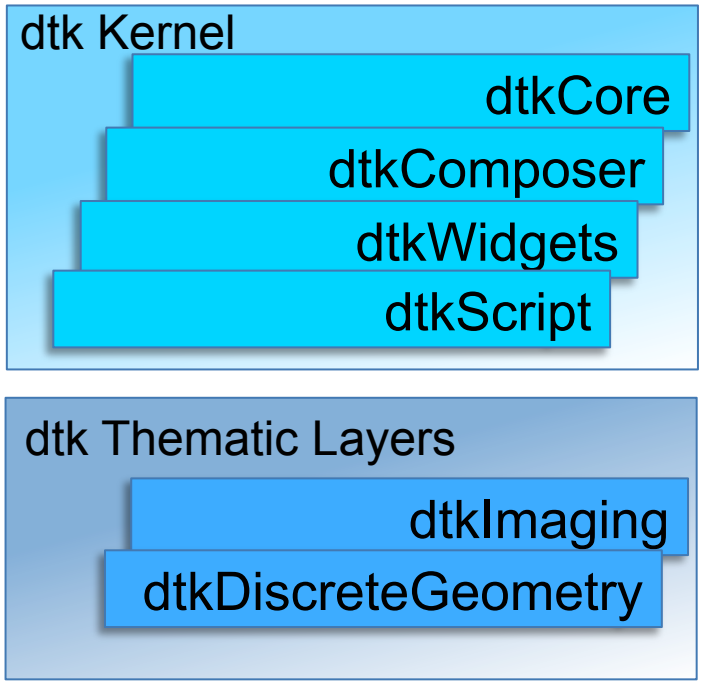
Plugin system
Visual Programming
GUI primitives
Wrapping tools
Distributed tools



Generic tools

dtk Thematic

Abstract Concepts for dedicated domains

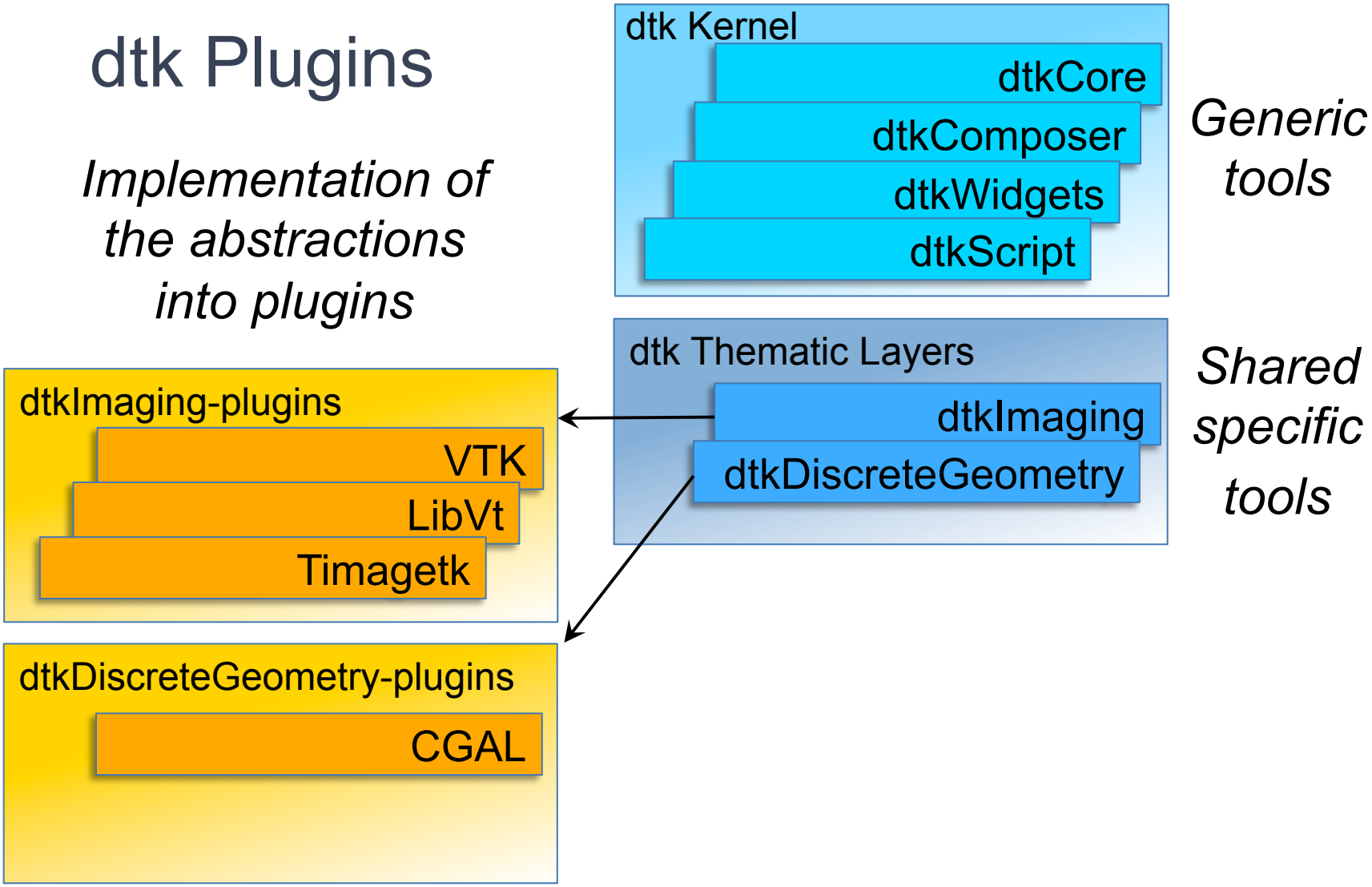


Generic tools

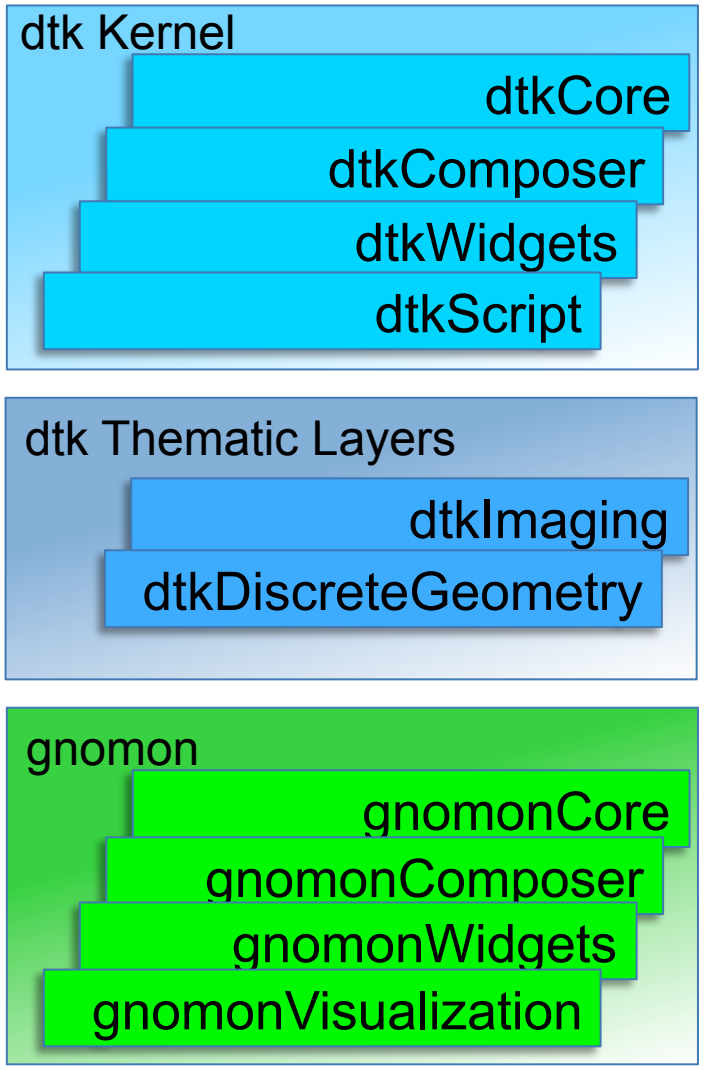
Shared specific tools

dtk Plugins

Implementation of the abstractions into plugins

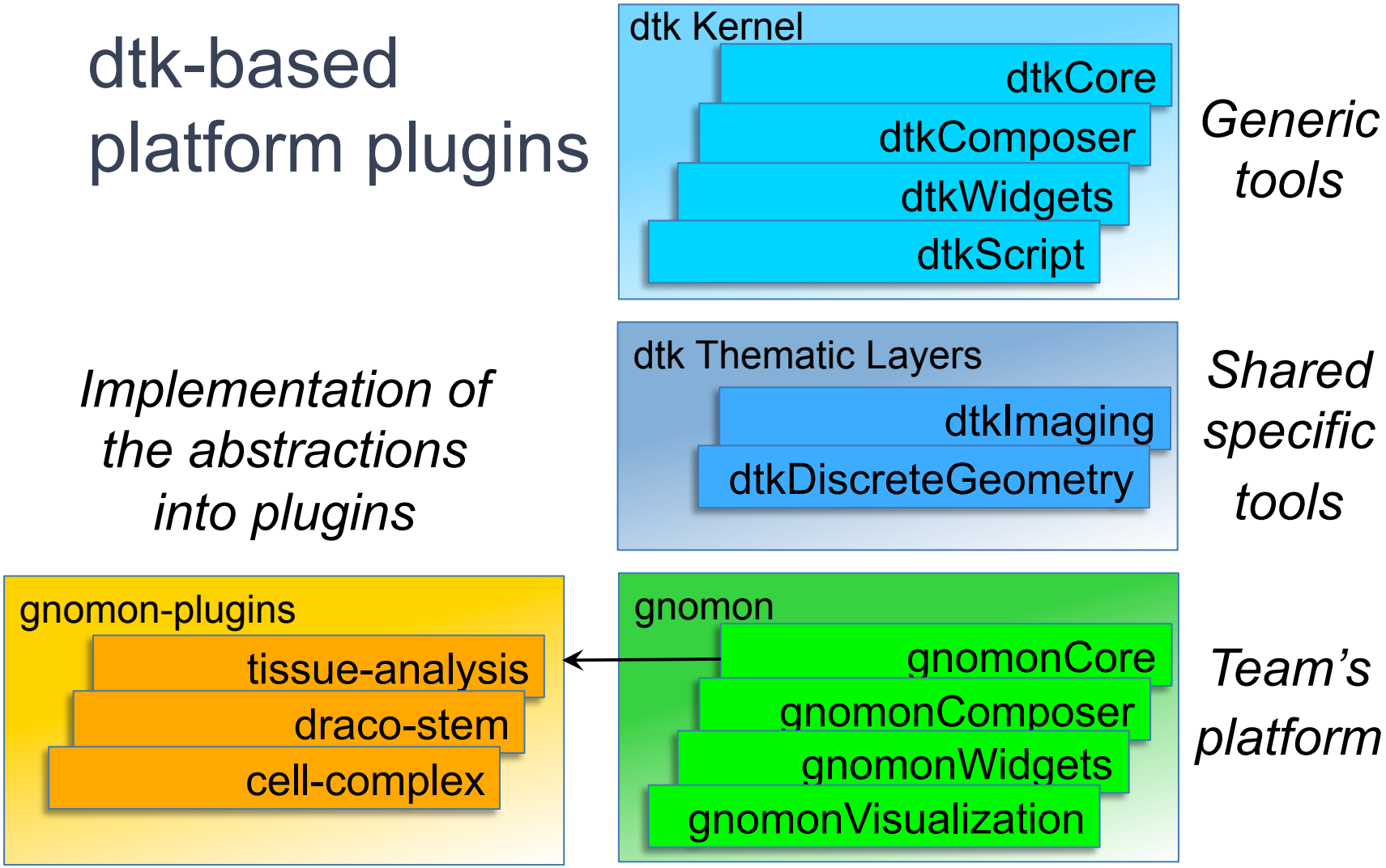


dtk-based platform



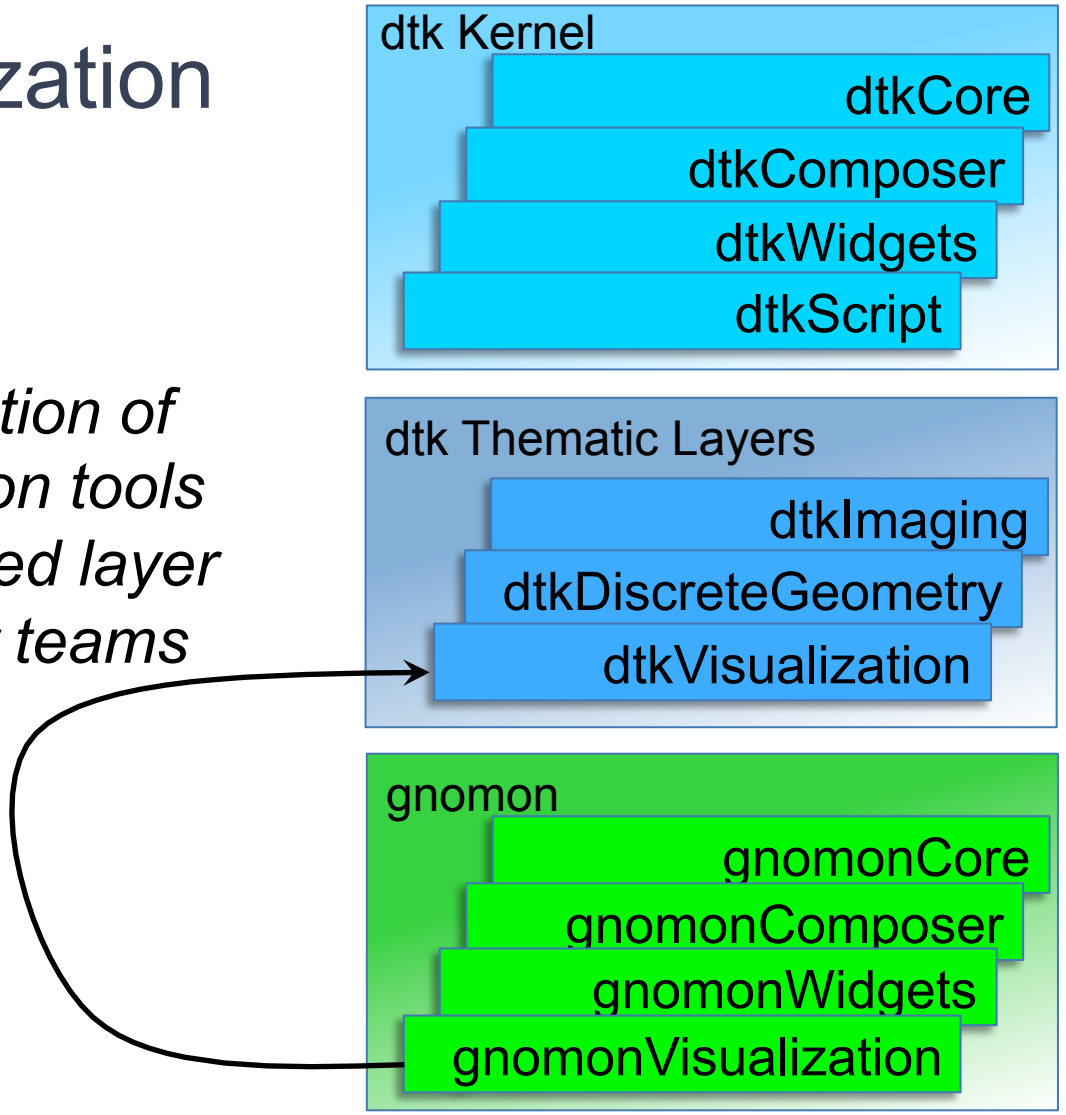
Abstract Concepts for team's domains

dtk-based platform plugins



Factorization

Factorization of visualization tools into a shared layer with other teams



Thank you!

Follow us on https://iww.inria.fr/sed-sophia/feed

inventors for the digital world