

Value-Driven Platform

The Platform is one of the keys to delivering continuous business value to customers, enabling fast-to-market capability and enhancing built-in-quality products. We explore some key principles for platform design and how to move the paradigm towards platform responsibility.

Giovanni Mariotta

Staff Architect

17-06-2022

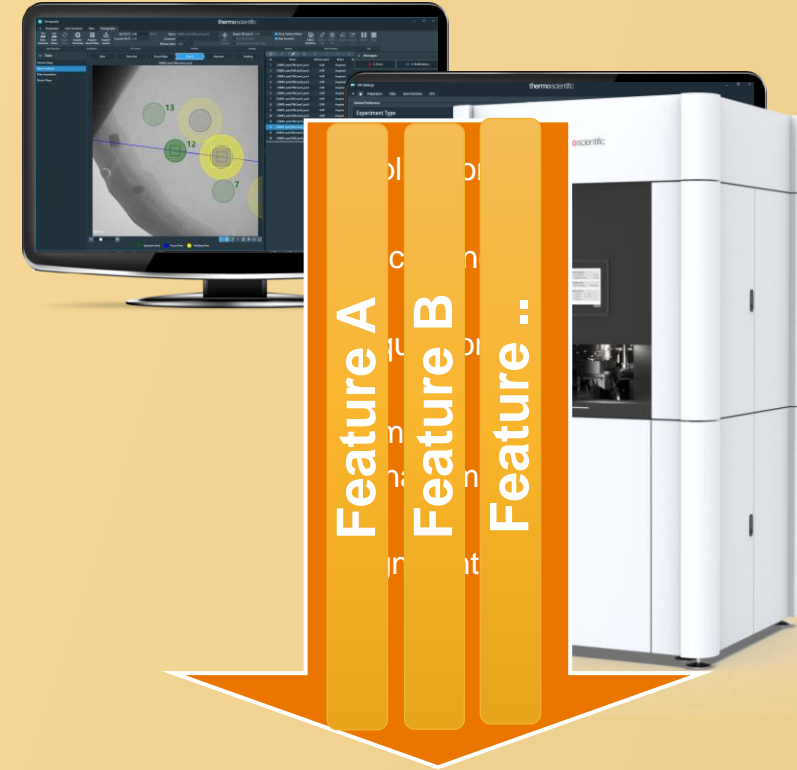
 The world leader in serving science



Get to know



- In Netherlands from 2014
- Working in **Thermofisher** from 2016 as Staff Architect
- Life Science ART System Architect
- 10 Teams distributed in Netherlands/France/Czech Republic



H O B B Y

Photography is one of my passion....
Macro is my favorite! 😊



Agenda

1 The Core Business

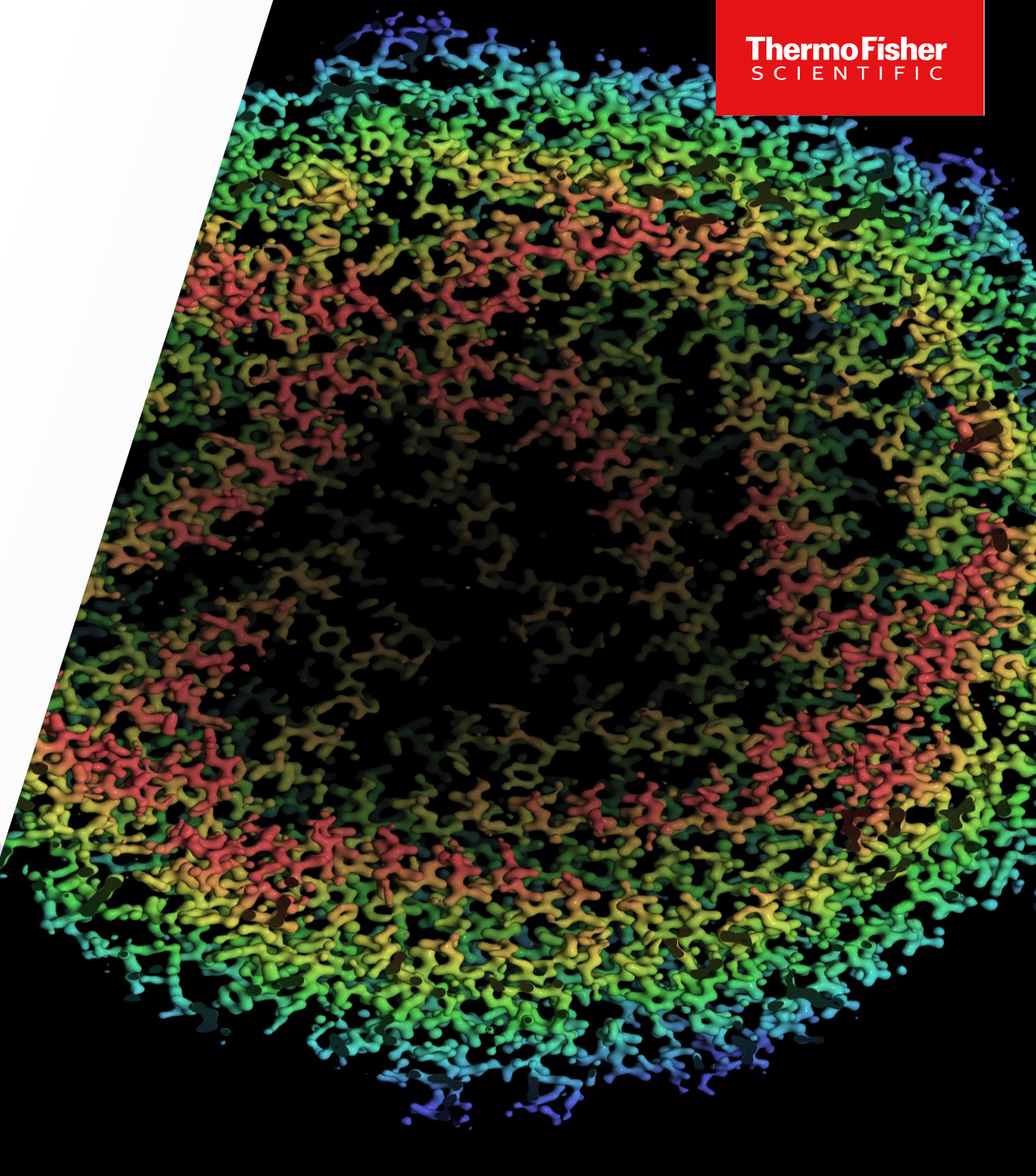
2 Reference Architecture supporting Customer Value

3 Value Driven Platform

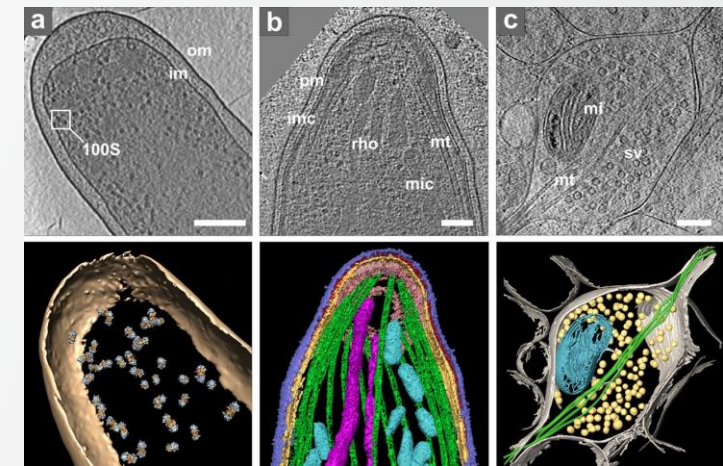
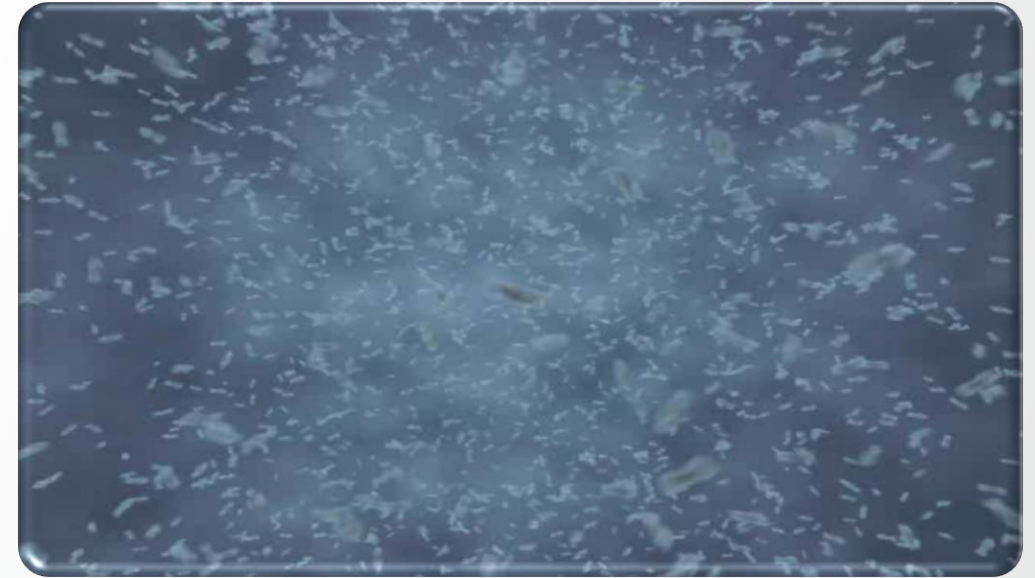
4 Conclusions



The Core Business



The Core Business



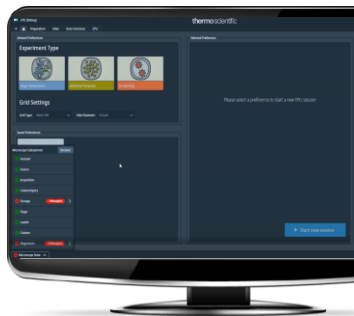
a) *E. Coli*
b) *Plasmodium* organelles and cytoskeleton
c) Presynaptic terminal
Lucic et.al., 2013, Journal of Cell Biology

The Core Business

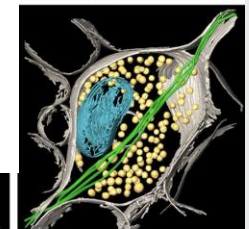
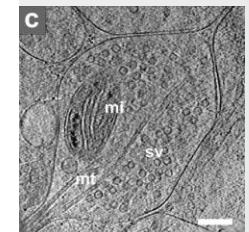
Jacques Dubochet, Joachim Frank and Richard Henderson share the prize for developing a technique to image biomolecules.



From left: Jacques Dubochet, Joachim Frank and Richard Henderson helped to develop cryo-electron microscopy.

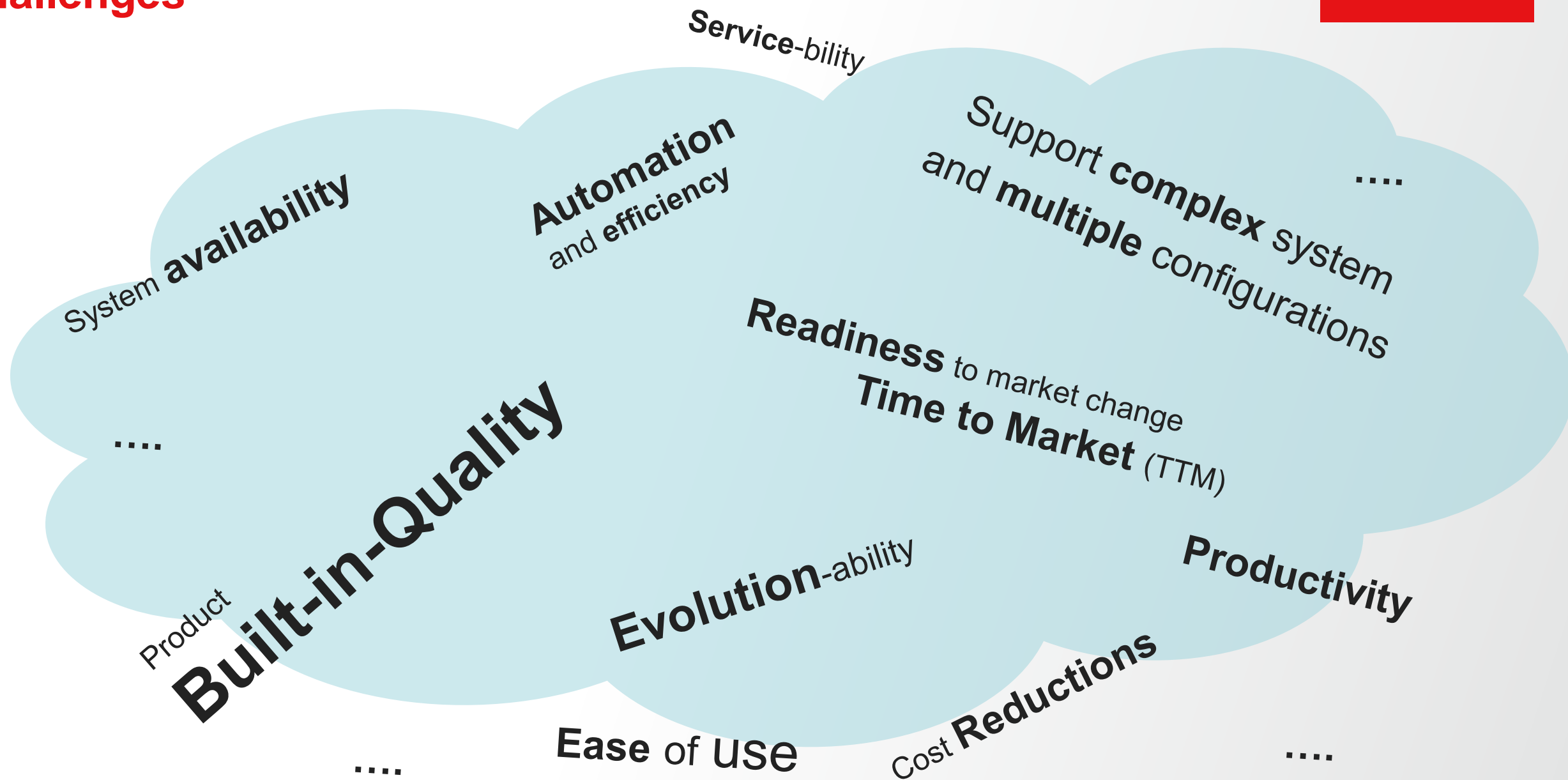


The 2017 Nobel Prize in Chemistry has been awarded for work that helps researchers see what biomolecules look like.



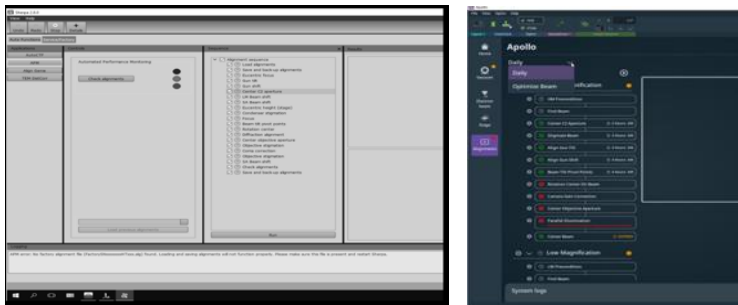
- a) *E. Coli*
 - b) *Plasmodium* organelles and cytoskeleton
 - c) *Presynaptic terminal*
- Lucic et.al., 2013, *Journal of Cell Biology*

Challenges



The Paradigm Shift

From application suite development...



- Different looks and feels
- Multiple teams dedicated to one product not connected each other's
- Not reusable libraries and issue
- No automation in testing

...To **develop Platform** to support **Customer Values**

- Reference Architecture based on customer value and reusable modules
- Vertical development and cross team interaction / connection
- Built-in-Quality product
- Fast to Market
- Same looks and feels
- ..



The Paradigm Shift

From application suite development...



“A **set of software** and a surrounding **ecosystem of resources** that helps to grow the business based on **customer values**”

“A platform **enables growth** through **connection**: its ability to connect external tools, teams, data, and processes.”

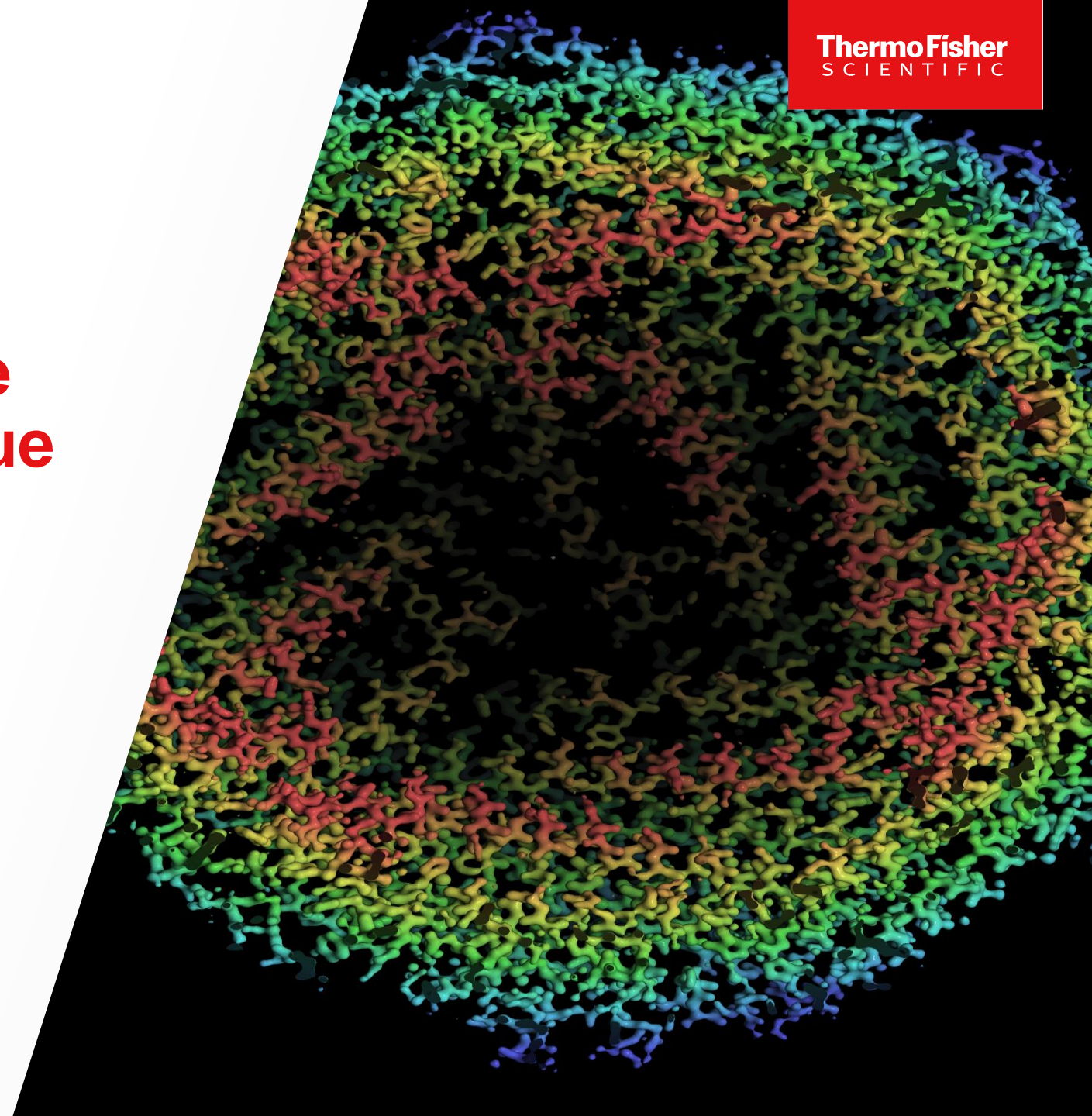
- Different lo
- Multiple te
- Not reusal
- No autom

- Reference Architecture based on customer value and reusable modules
- Vertical development and cross team interaction / connection
- Built-in-Quality product
- Fast to Market
- Same looks and feels
- ..



Values

Reference Architecture driven by Customer Value



“Platformization” into Practices: Customer Value First

Customer Value

Workflow analysis – maximize customer value

What customer needs

- Identify Customer Value
- Identify Critical To Quality (CTQ) Parameters
- Relate CTQ Parameters to Features



Automation



Ease of Use

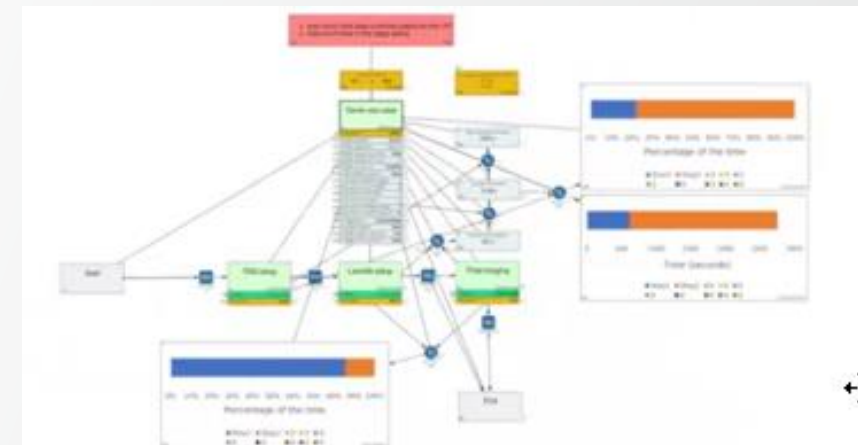
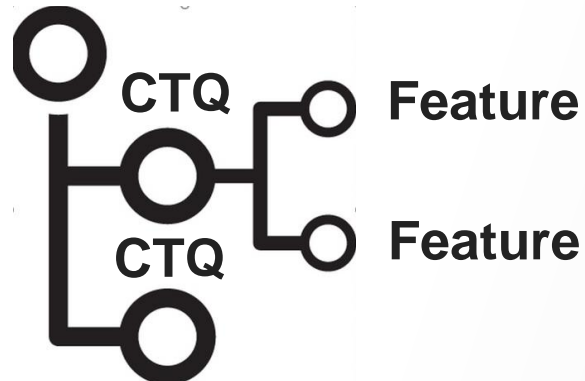


Throughput

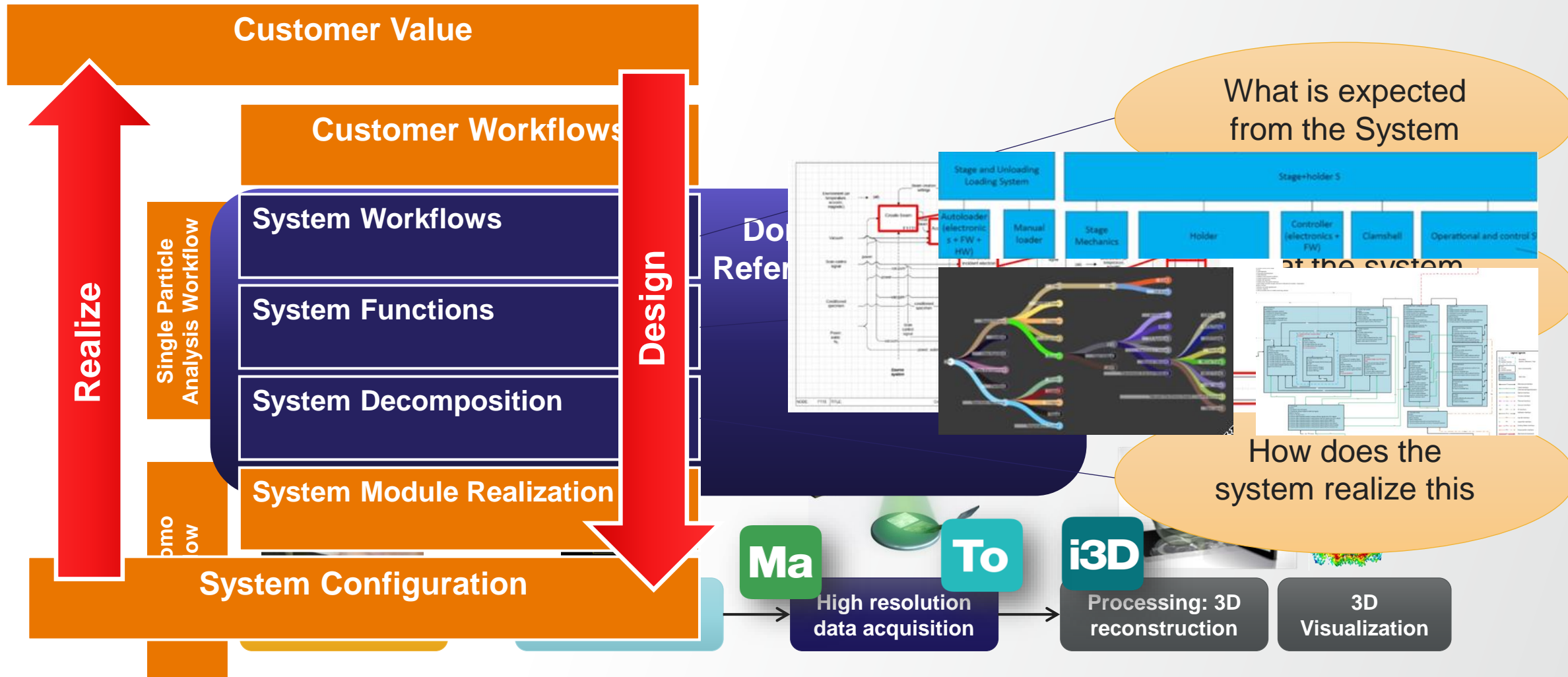


System Availability

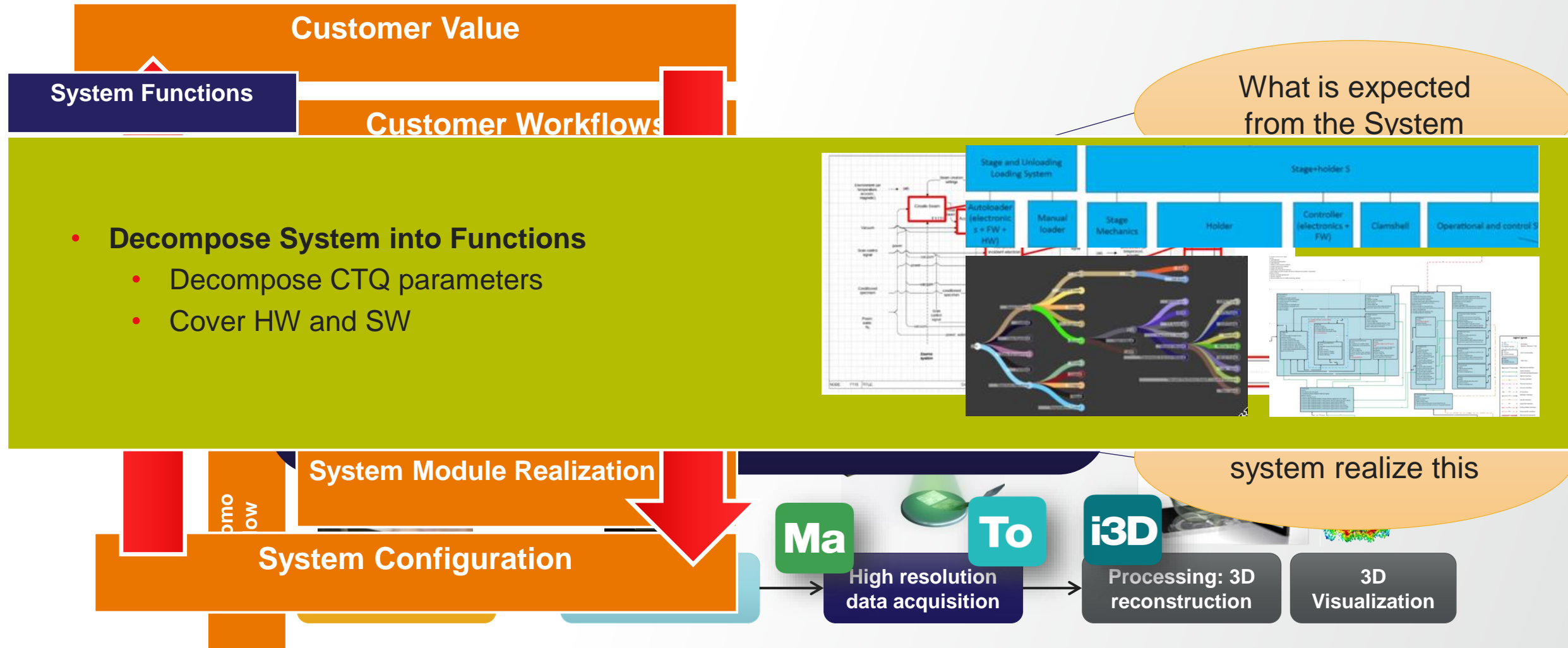
Value



“Platformization” into Practices: Customer Workflows



“Platformization” into Practices: Customer Workflows



“Platformization” into Practices: Customer Workflows

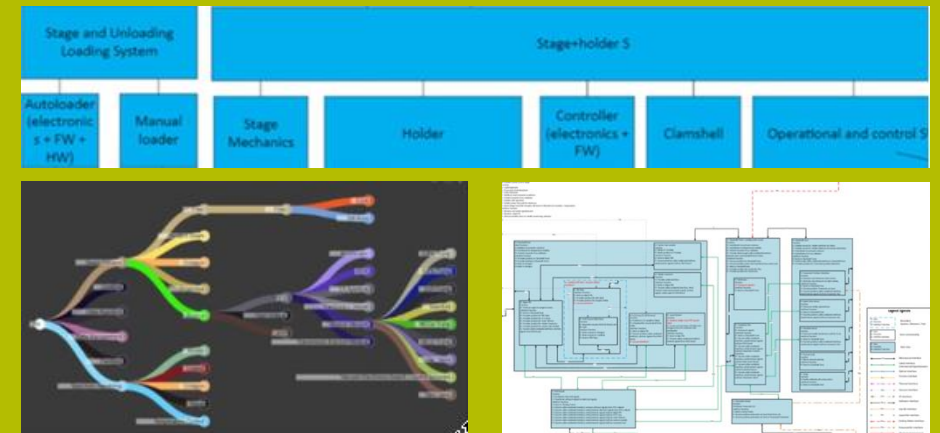
Customer Value

System Decomposition

Customer Workflows

What is expected from the System

- **Decompose System into Modules**
 - Allocate Functions to Modules
 - Logical grouping of functions
 - Transfer Functions definition for Modules (based on Function CTQ Parameters)
 - Boundary Diagrams: identify interfaces between modules
 - Compatibility
 - Risk Management (FMEA)



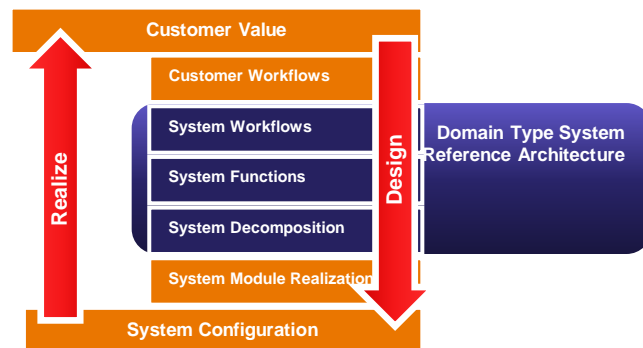
System Module Realization

System Configuration

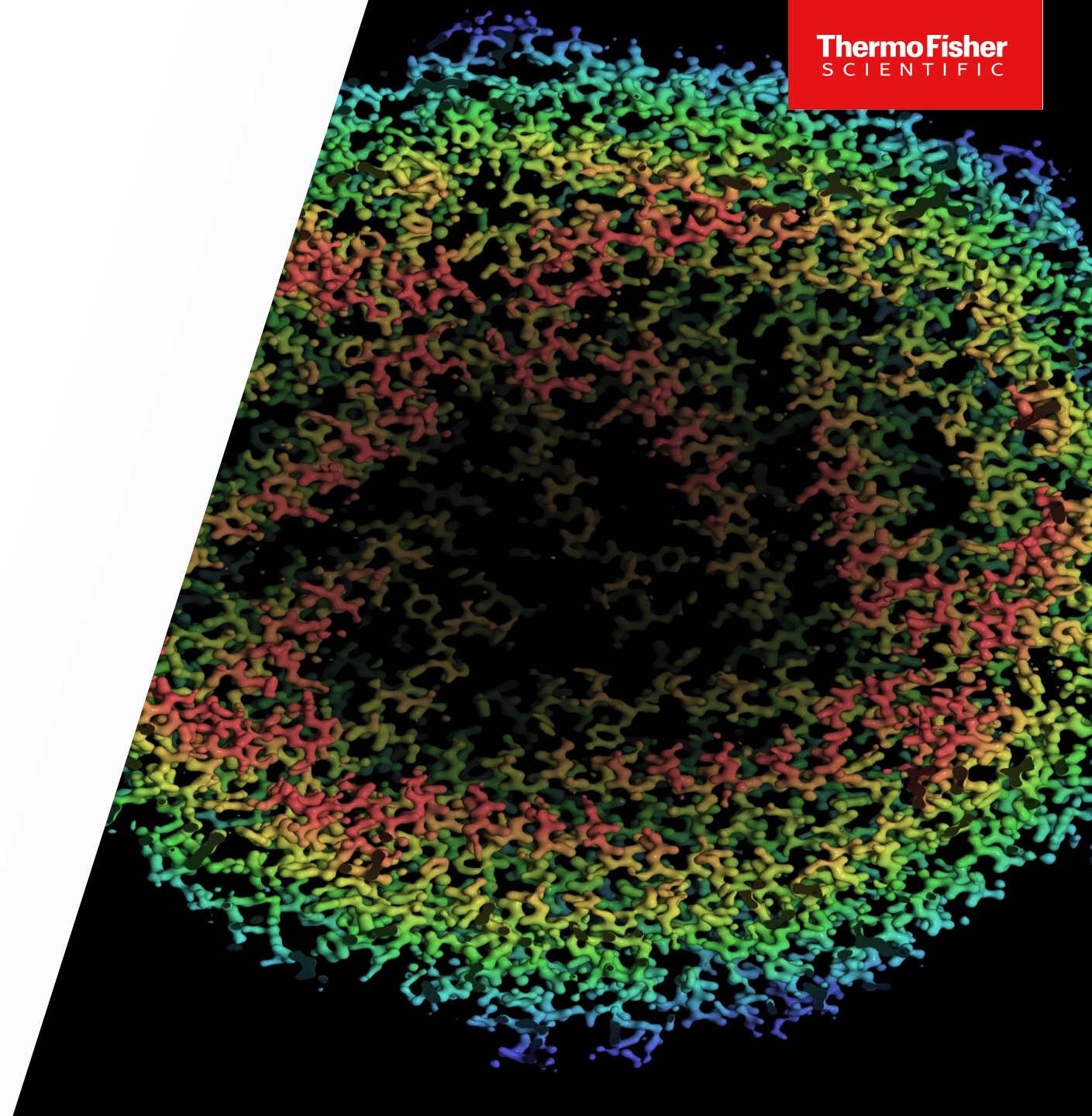
system realize this



With Multiple Use cases and workflow how to efficiently develop, share capabilities and gain time to market with customer value?



Value Driven Platform



Scaling up on Platform level

Add shared Capabilities to Platform associated to business value and support multiple use cases

- **Microscope Control Platform**

Allow applications to work together seamlessly, without workarounds and automate integrations during development and deliver customer value by connecting to Microscope interface

Ease of use
Calibrations

TTM
Interact with Microscope for fast prototype

TTM and readiness to market change
Fast Application prototype and support innovation

Built-in-Quality
Test Automation

Ease of use
Alignments

Time to reconstruction
Image Quality
Automated algorithms

System Availability
Logging and Troubleshooting

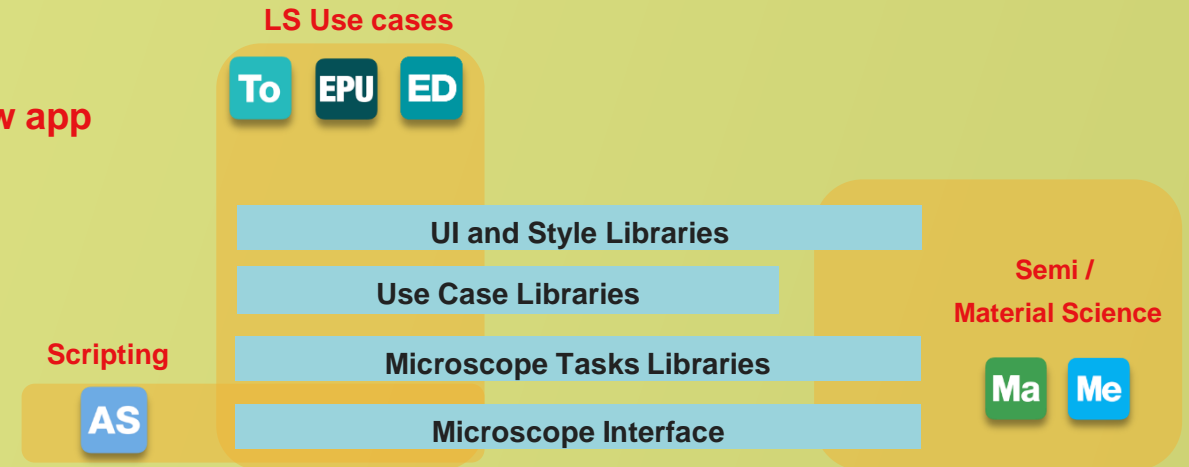
System Availability
Preventive Maintenance System Profile

Microscope Control Platform

Some Examples

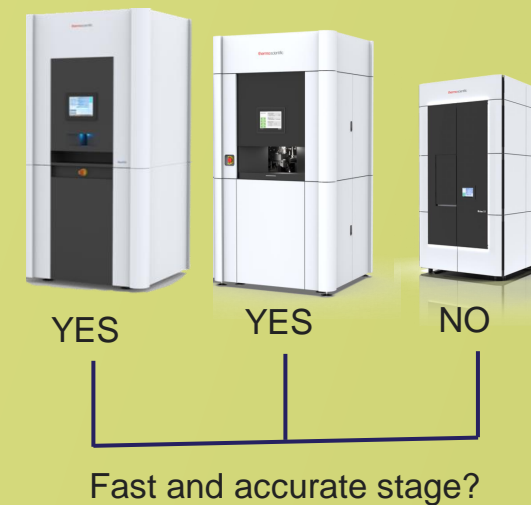
Support multiple use cases – Platform to build and prototype fast new app

- Platform provides common and shared libraries for fast prototype and cover different use cases



Use Microscope Interface by support multiple configurations

- Clients' agnostic from system configuration: capabilities-based interface

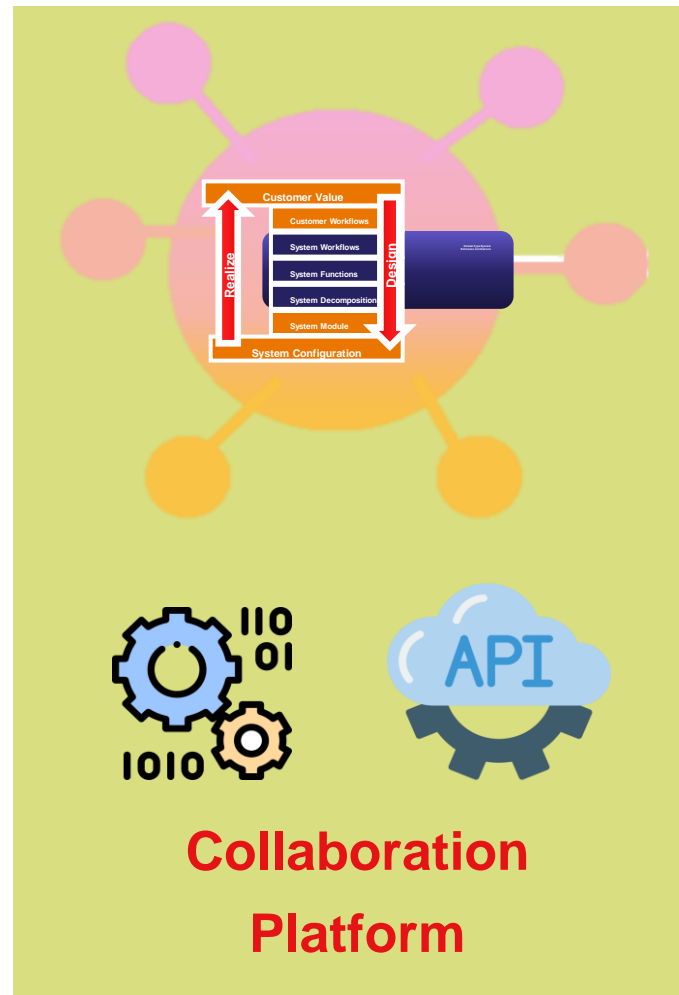


EPU To ... Me

Scaling up on Platform level

- **Collaboration Platform**

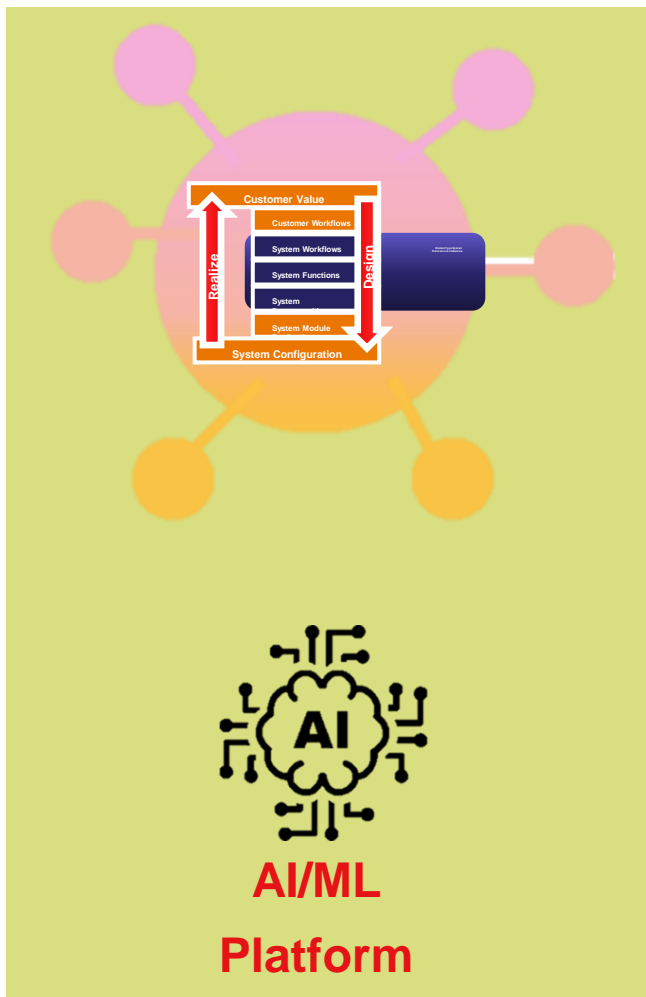
Support value delivered by customer workflow with dedicated processing tools and plugins created by community and internal developers



- **Practices**

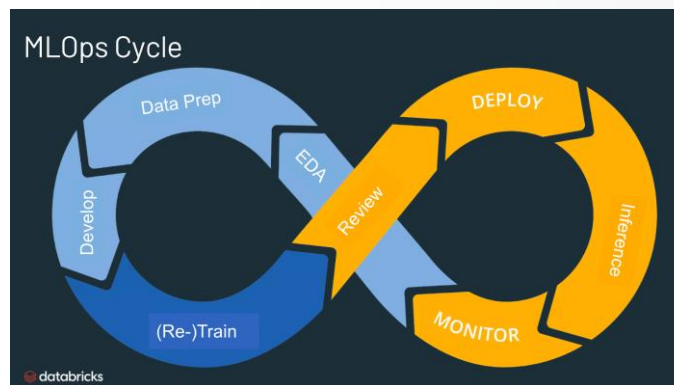
- Pay attention to expose API decoupled by internal technologies
- Provide SDK and example for collaboration
- Reuse internal services and capabilities as part of the platform (i.e security, logging, profile)

Platform for AI/ML Capabilities... maybe in future?



- **Platform for AI/ML Capabilities**

Support dedicated AI/ML capabilities for development and production to deliver business value by expanding capabilities on Artificial Intelligence, as development and as infrastructure



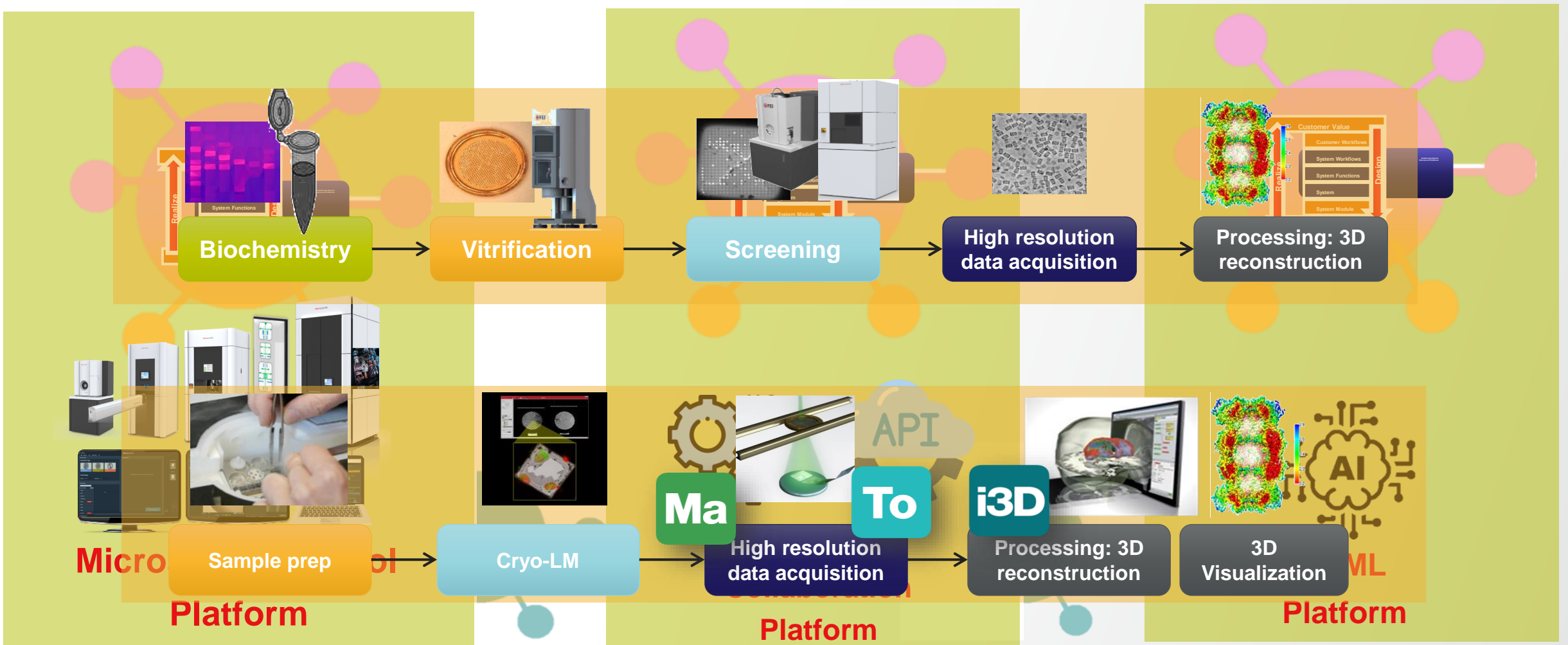
TTM
Fast Prototype
Model Creation

TTM: Dev Efficiency
Model Repository and
Production deployment

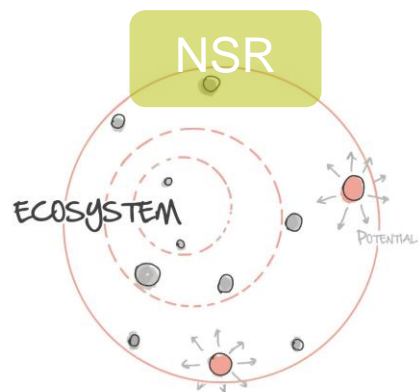


An Ecosystem of dedicated Platforms

“A set of **software** and a surrounding **ecosystem of resources** that helps to grow the business based on customer values”



Principles for Platform Design



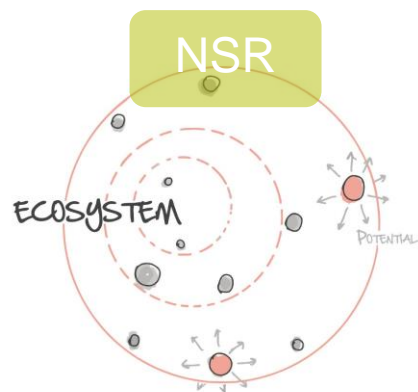
Recognize the potential that
grows at the Edge

Recognize the potential that grows at the Edge

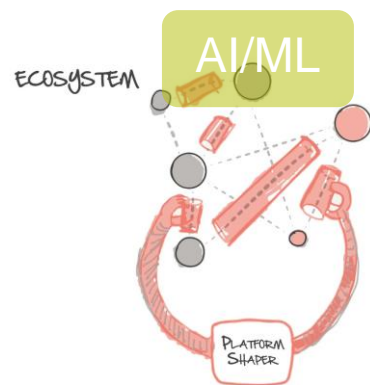
Support Not Standard Request (NSR)

Apple didn't hire developers but helped them to distribute their software

Principles for Platform Design



Recognize the potential that grows at the Edge

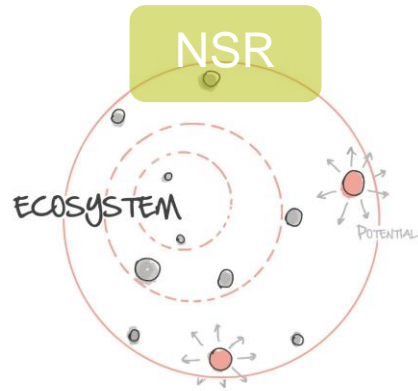


Design For Emergence

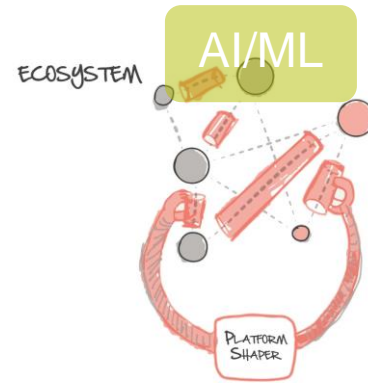
Design For Emergence

Platform strategies need to be designed to help an existing ecosystem to emerge, thrive and work better

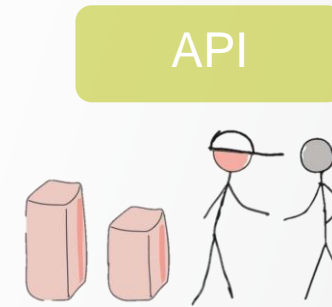
Principles for Platform Design



Recognize the potential that grows at the Edge



Design For Emergence

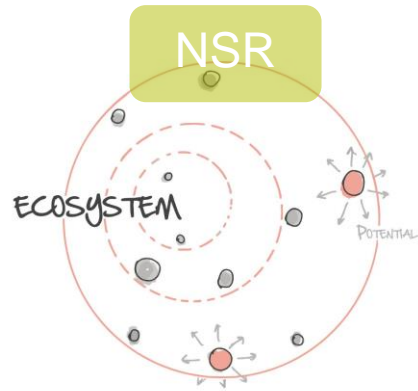


Use Self Organization to provide Mass Customization

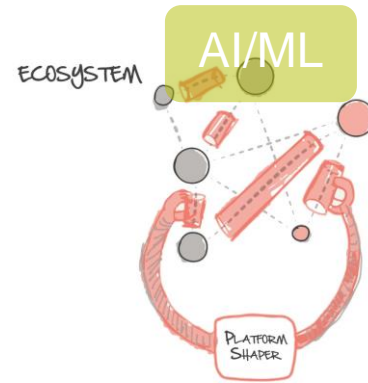
Use Self Organization to Provide Mass Customization

consumers demand personalized solutions to specific, custom, contextual and personal expectations. There's **no way to industrially generate mass customization**, this can only be achieved by helping entities in the ecosystem to *self-organize* around the potential
 Apple didn't plan which set of applications were to be developed. It let developers to grow a business out of their developers

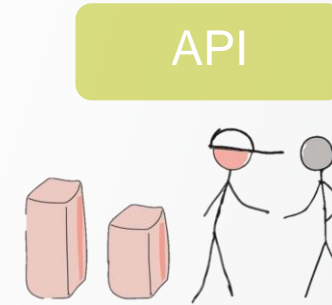
Principles for Platform Design



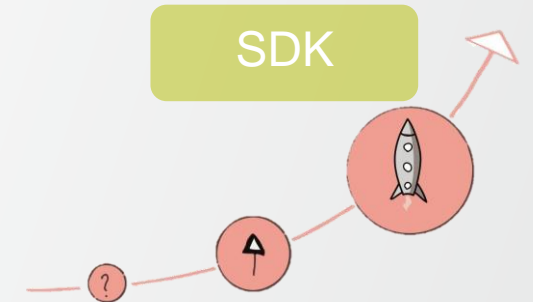
Recognize the potential that grows at the Edge



Design For Emergence



Use Self Organization to provide Mass Customization



Enable Continuous Learning

Enable Continuous Learning

Modern organizations need to offer participants a promise of *accelerated learning*. Their message needs to be: “*if you join us on the new terms of collaboration (platform), you’re going to learn faster than outside*”. When offering learning opportunities, the journey must be playful, and *flow-enabled*: participants need to learn step-by-step, first how to interact (*onboard*), then how to scale and manage scaled interactions (*get better*), then how to test themselves in new opportunities, new contexts.

Principles for Platform Design

Design For Disobedience

To let the players play at the edge of the rules: if a new, emergent, behavior appears as recurrent, then the platform designer might consider to be there to **institutionalize** it, making it a **feature**.

First let the ecosystem **dictate new expectations**, then **institutionalize them** and leverage them (grow them at scale), within time **make them components** that the ecosystem can use, to invent something new.

Recognize the potential that grows at the Edge

Design For Emergence

Use Self Organization to provide Mass Customization

Enable Continuous Learning



Design For Disobedience

Principles for Platform Design

NSR

ECOSYSTEM

AI/ML

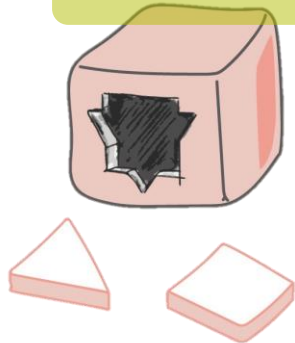
API

SDK

Design For Interconnectedness

This brings the organization to move beyond the very concept of a *customer* (somebody you *design a solution for*), and embrace the concept of **relationships**, and **interactions** between peers as the key element of business. When we design to facilitate interactions between producers and consumers of value, we need to intentionally design *with*, and *for* both parties.

NSR



Design For Disobedience

API

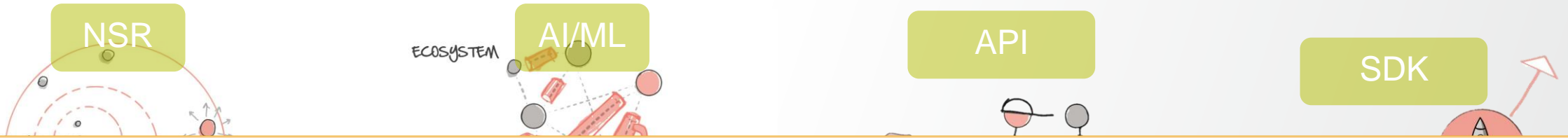
SDK

Collaborative
Tools



Design For Interconnectedness

Principles for Platform Design

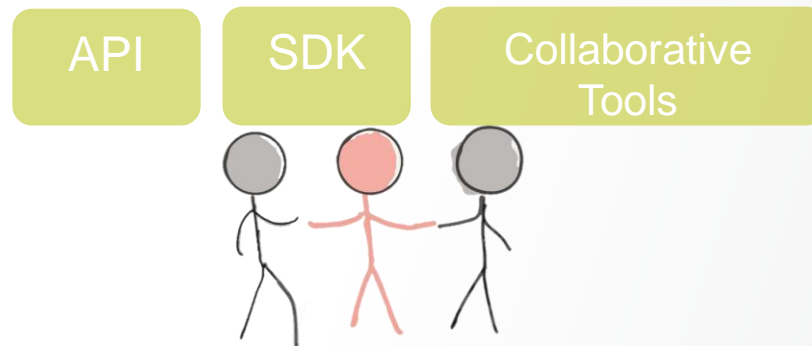


Identify with the Whole

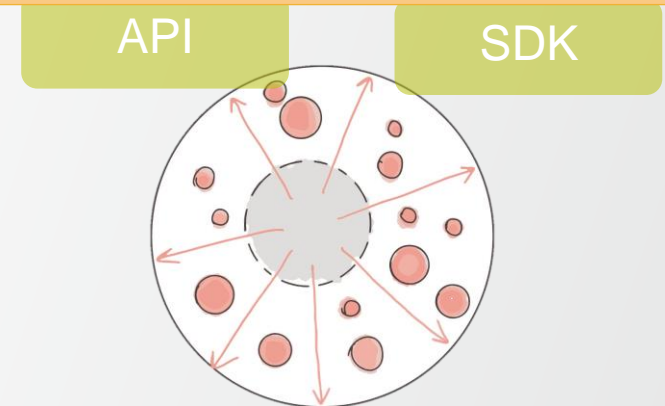
If the Premise is to grow a bigger market, this will eventually need to be less controlled from the organizer and the organizer will need to identify more with the whole and think strategies that actualize the opportunity for the whole instead of the organizer alone.



Design For Disobedience



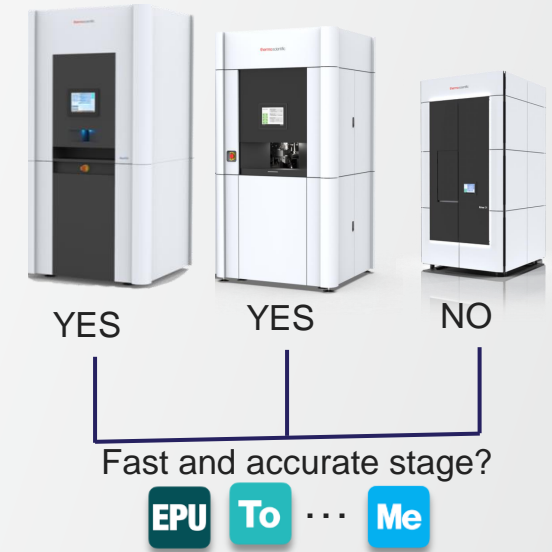
Design For Interconnectedness



Let go the identity, identify with the whole

Other food for thoughts

- Use **Capability based interfaces** to avoid coupling with configuration, module and platforms capabilities

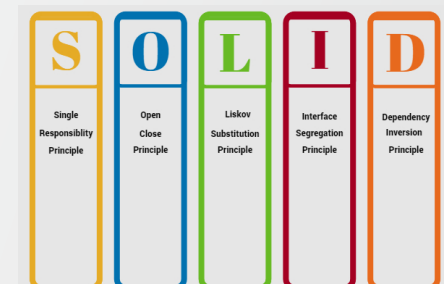


- Use **off-the-box technologies**

- Try to have common challenges and not based on your custom framework



- Give a **clear scope** to your Platform: **SOLID** principles also at platform level



Thank you

