BioDynaMo

Biology Dynamics Modeller

IT Technical Forum 01.12.2017

Lukas Breitwieser and Ahmad Hesam

CERN openlab

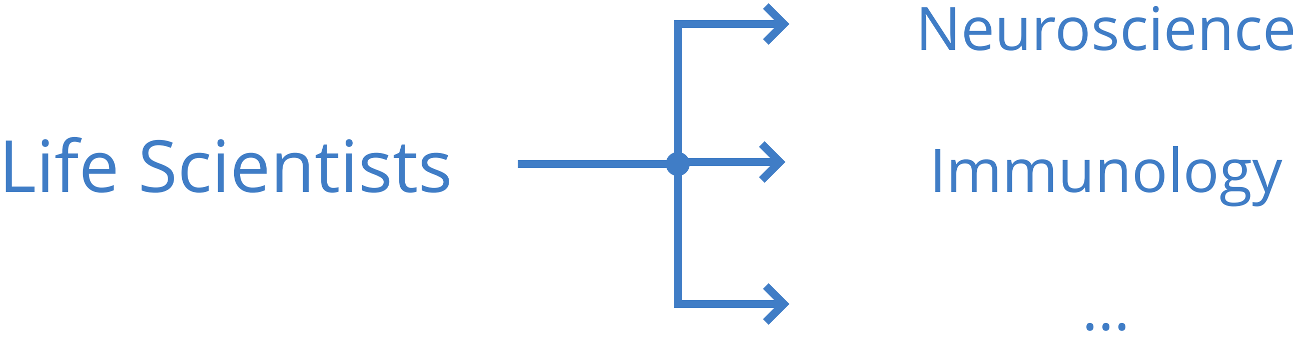
CERN   CERN openlab   Newcastle University   Kazan Federal University   Innopolis University   intel

Outline

- Background
- Platform
- Future Challenges
- Summary

CERN openlab

Background

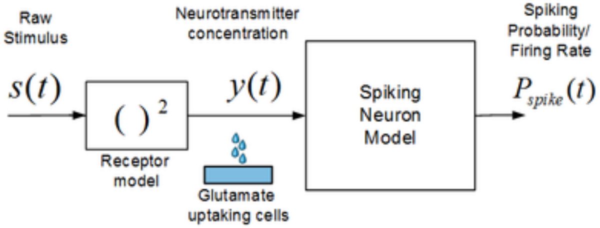


Life Scientists

Neuroscience

Immunology

...



http://hitconsultant.net/2016/09/26/infographic-future-personalized-medicine/

The Problem

More models

- Higher complexity
- Larger scale

How to simulate fast?

Life scientists often make simulations

specifically for their own models

Messy!

mech

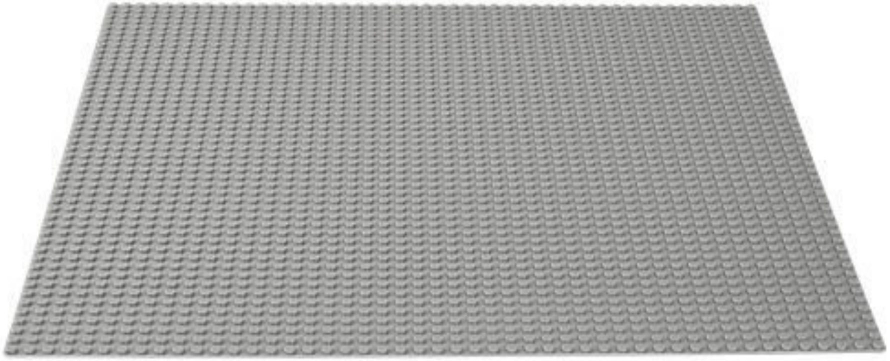
Ce

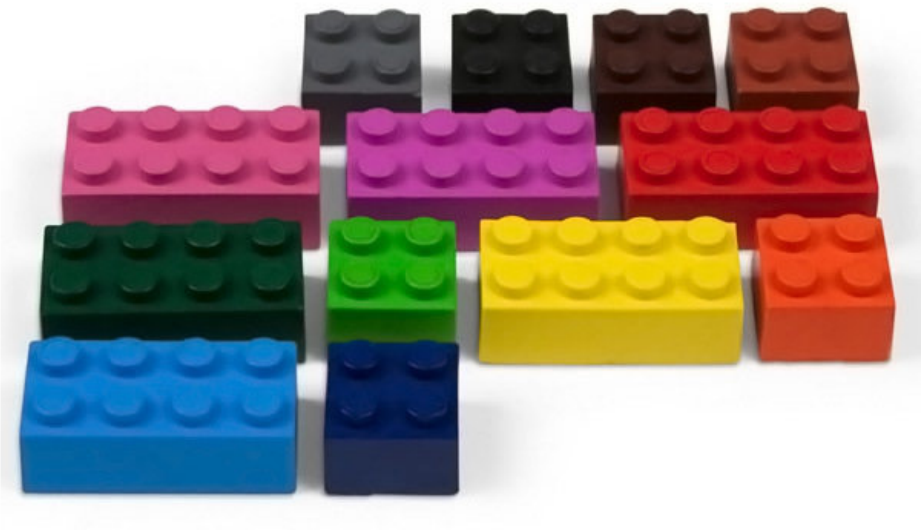
or

http://www.owlicious.com/activities/emda17vs/2013/4/26/homemade-play-dough-monsters
http://www.momtastic.com/images/stories/parenting/0212/120229-Finger-Painting.jpg

BioDynaMo

Biology Dynamic Modeler

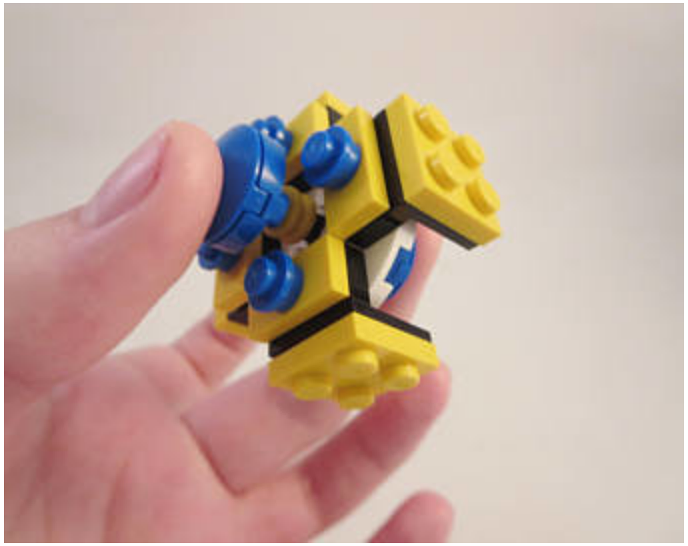




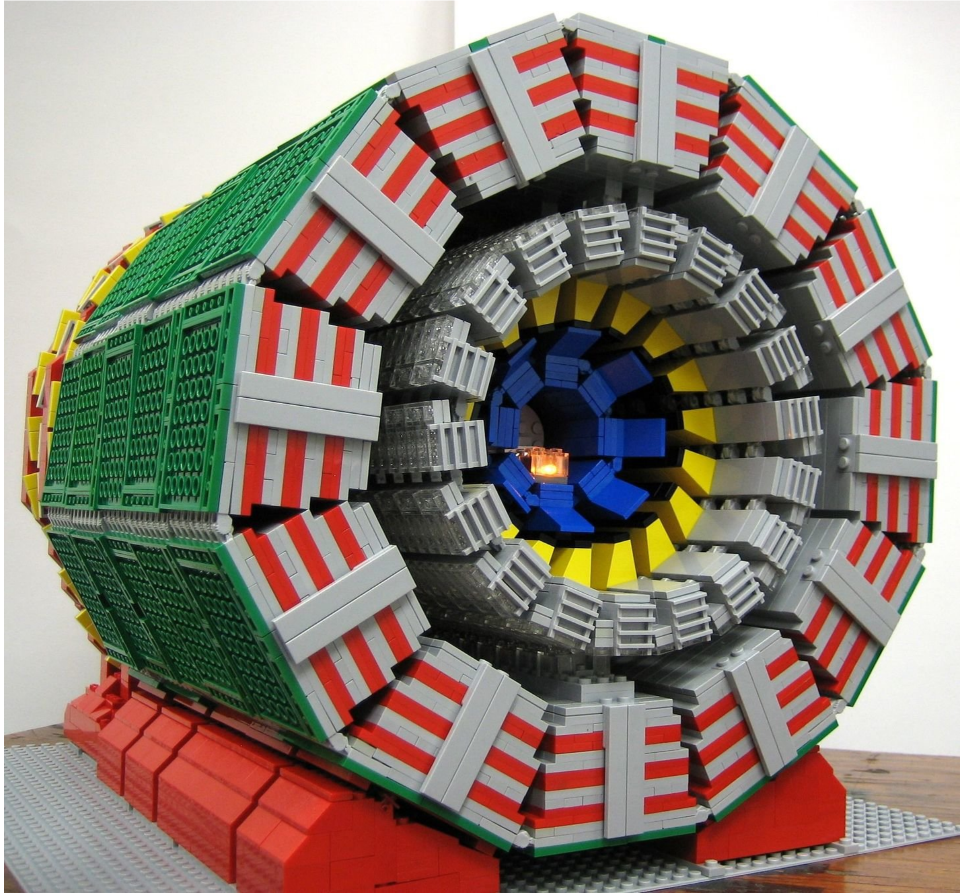
Modularity

https://shop.lego.com/en-US/Gray-Baseplate-10701
https://www.geekalerts.com/stack-a-doodle-stackable-lego-block-crayons/

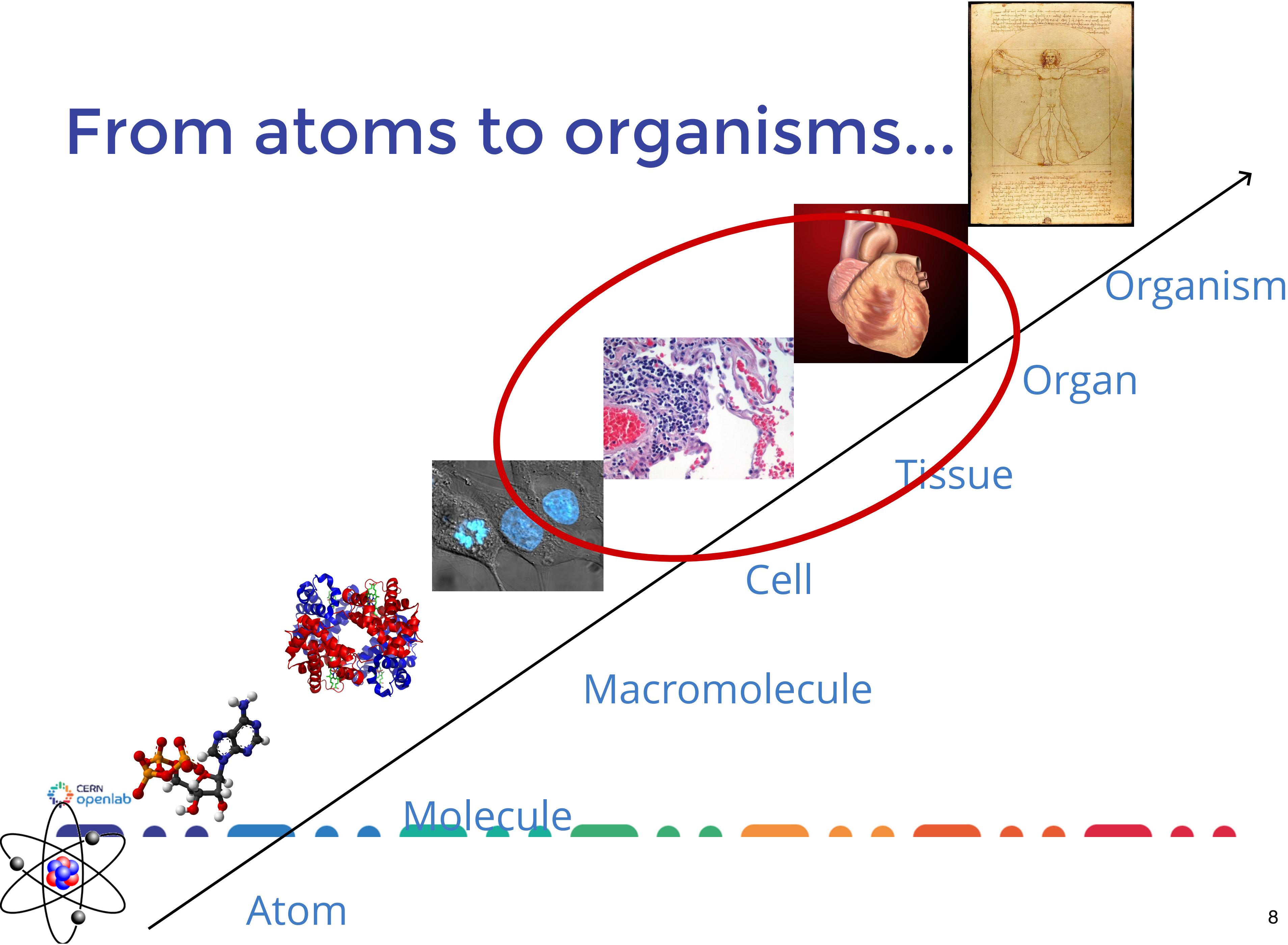
Create, extend



Share…



From atoms to organisms...



Atom

Molecule

Macromolecule

Cell

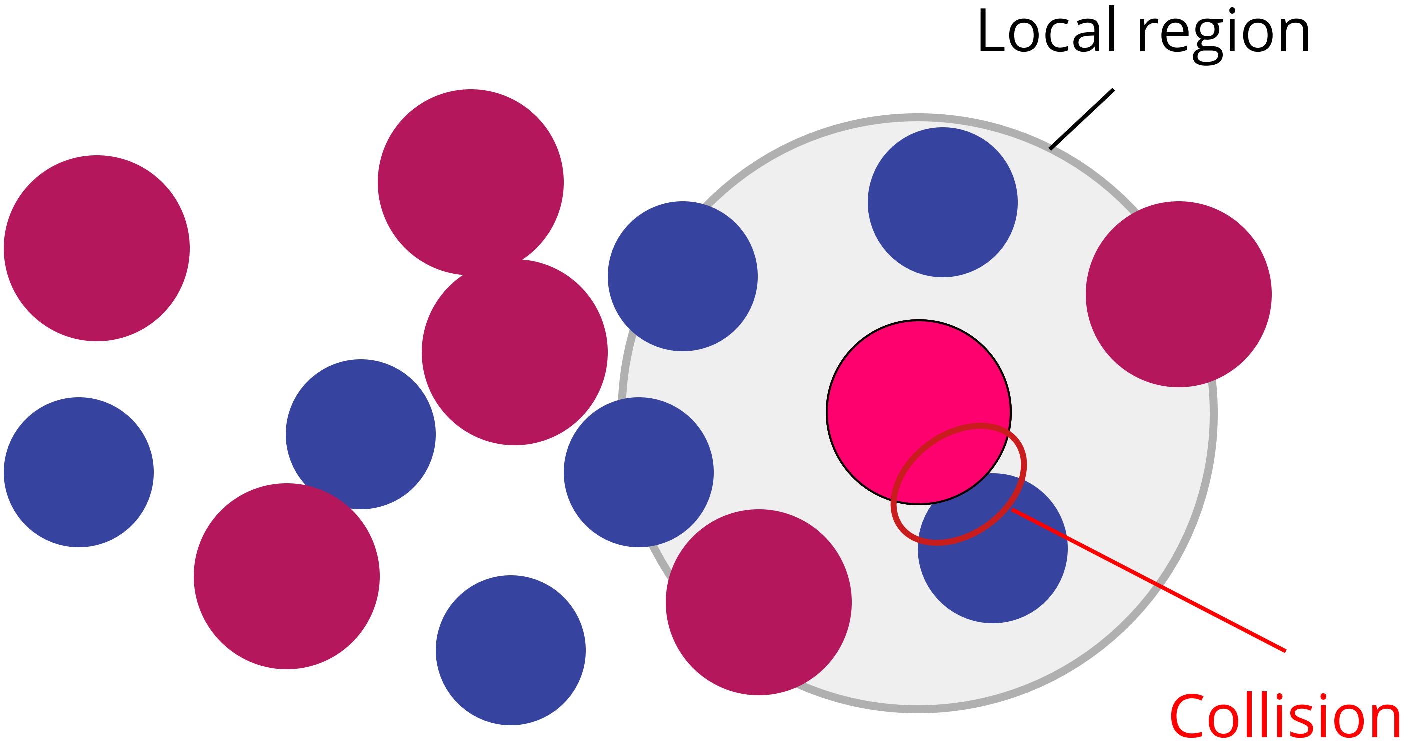
Tissue

Organ

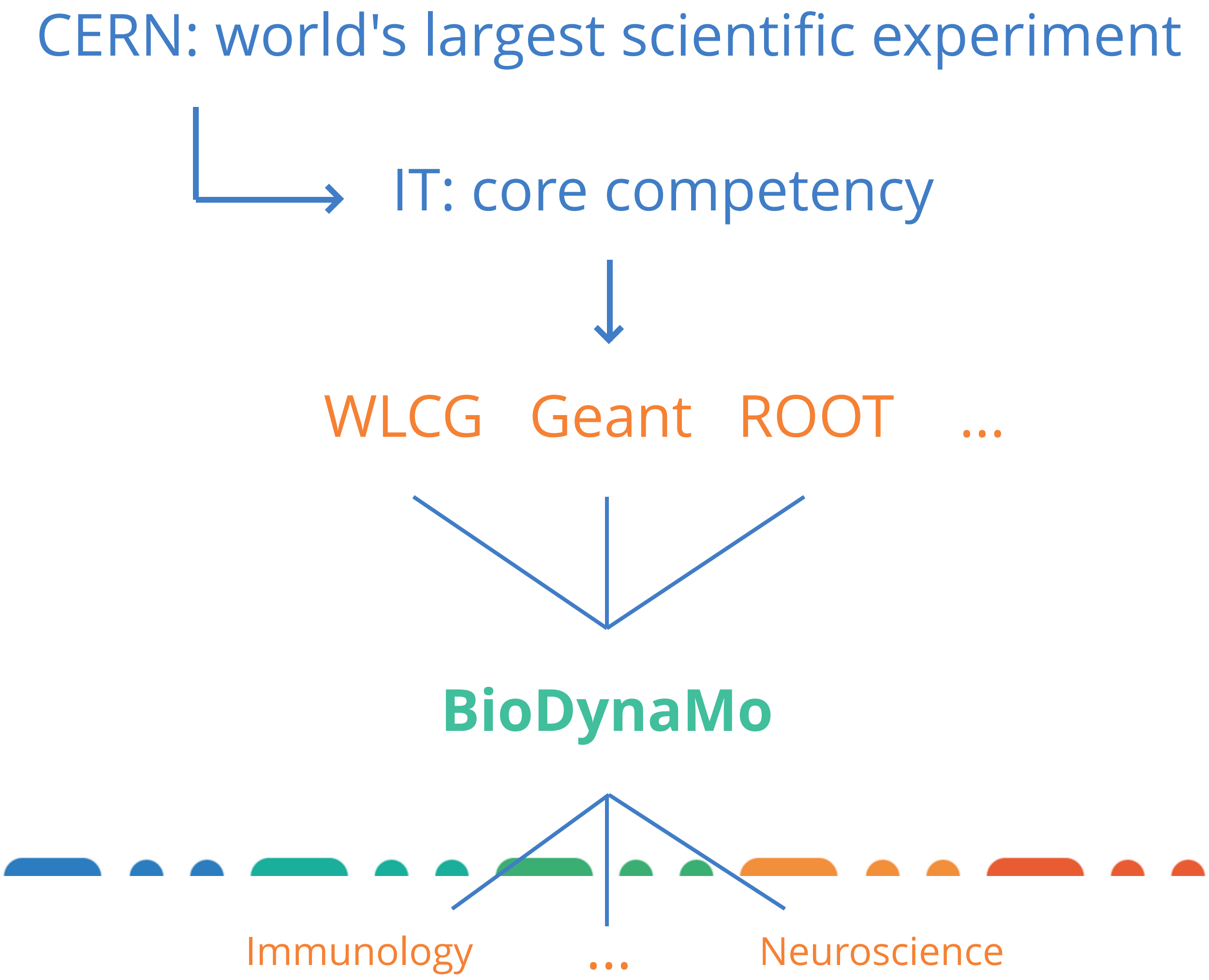
Organism

Agent-based simulations

Simulation object = *Agent*



Why at CERN?



CERN: world's largest scientific experiment

IT: core competency

WLCG Geant ROOT ...

BioDynaMo

Immunology ... Neuroscience

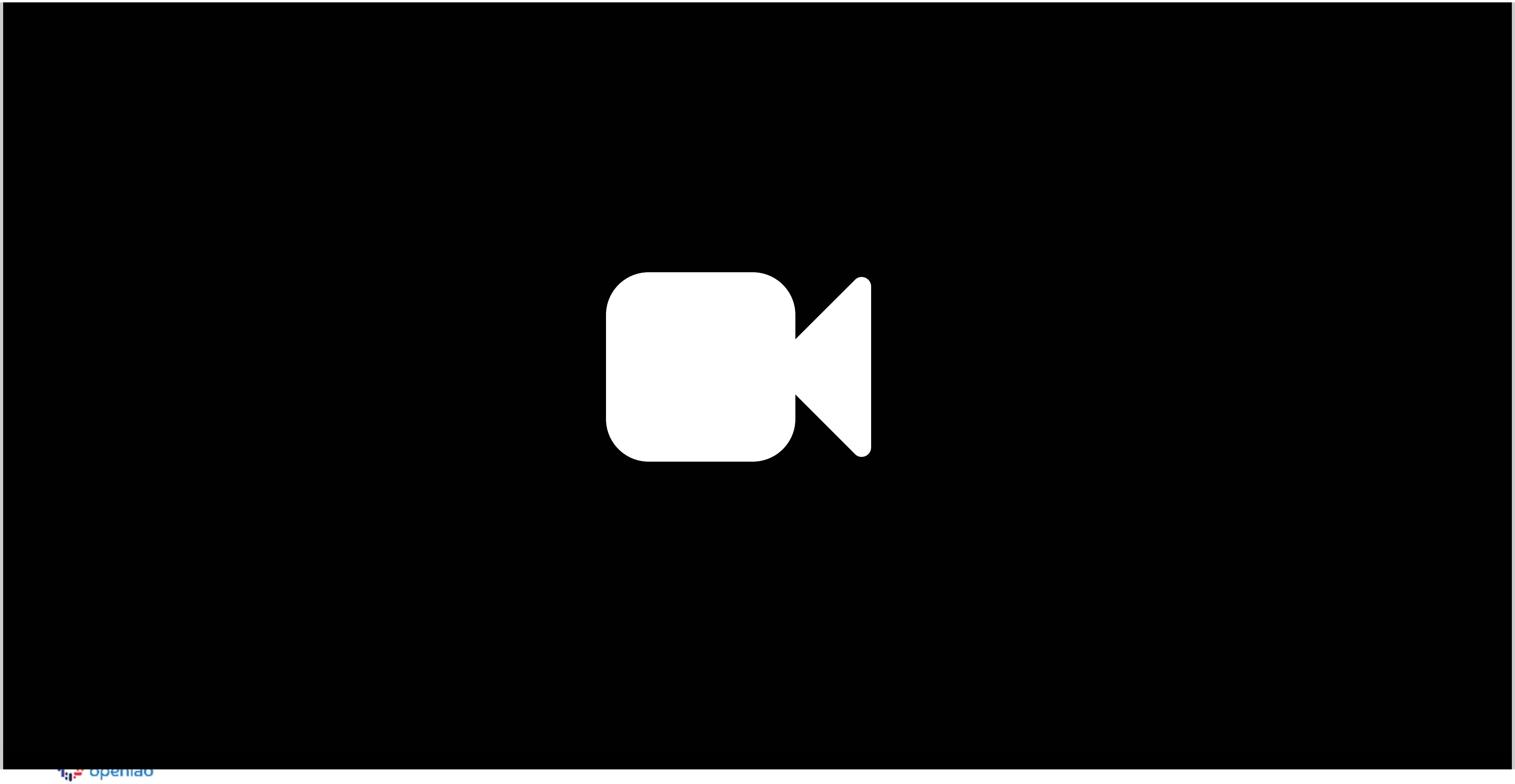
CERN openlab

Current BioDynaMo use cases

Newcastle University PhD Student: Jean de Montigny

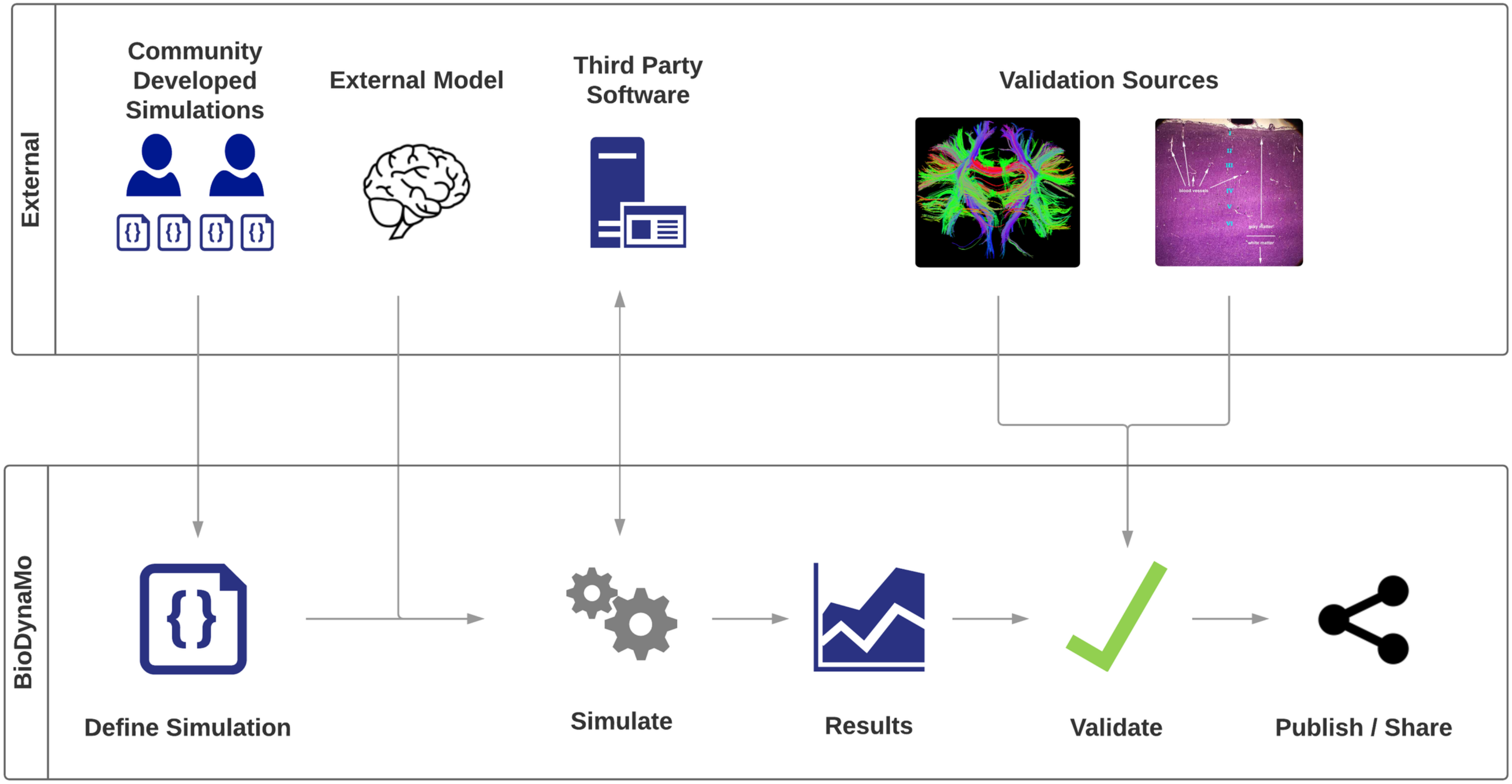
Tumor growth

Retinal mosaic formation



Platform

WORKFLOW DIAGRAM



External

Community Developed Simulations

External Model

Third Party Software

Validation Sources

BioDynaMo

Define Simulation

Simulate

Results

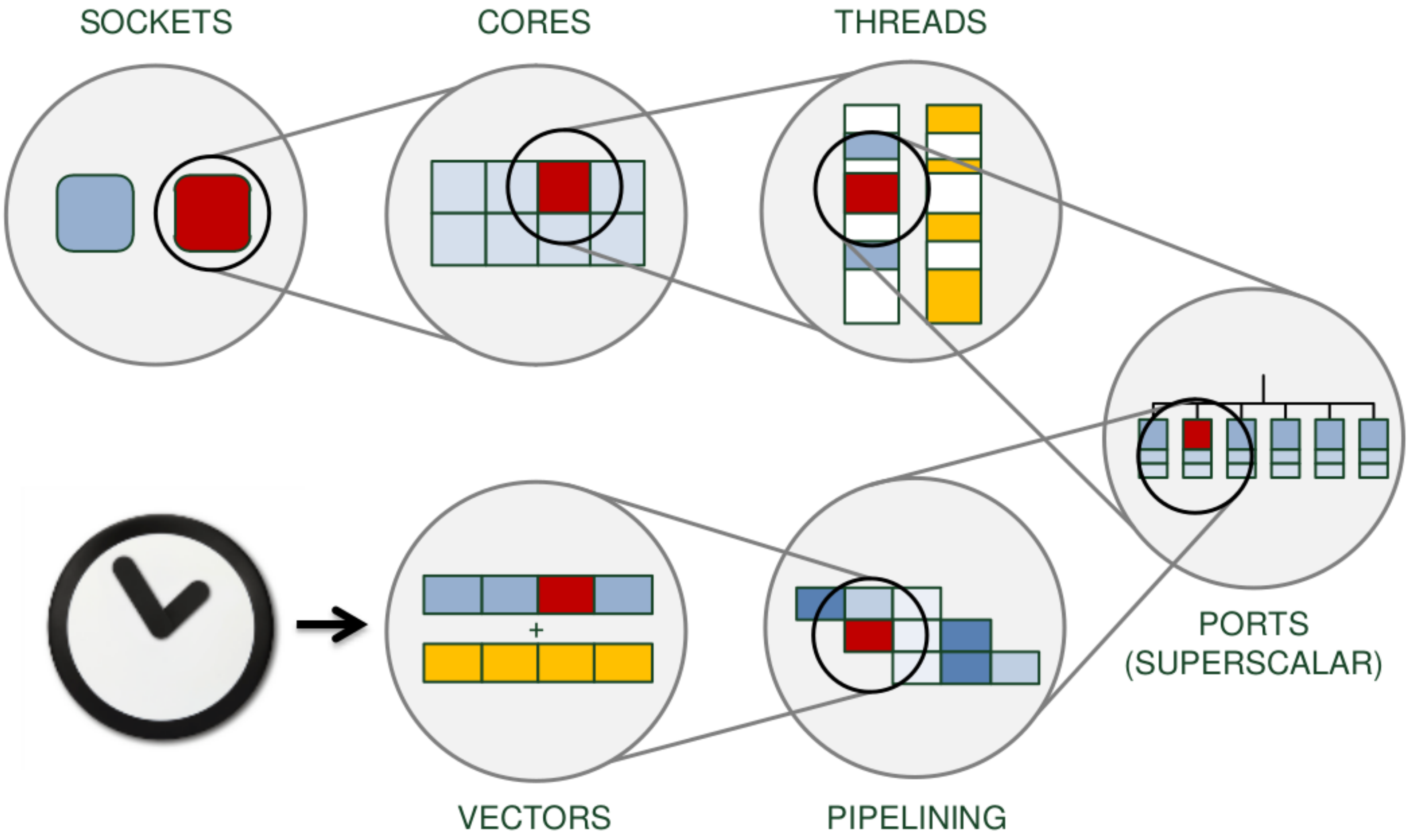
Validate

Publish / Share

Large-scale

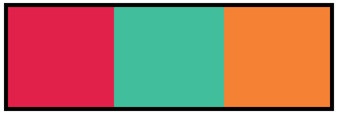
- Efficient use of modern hardware
- Cloud / distributed Runtime

Multiple Level of Parallelism

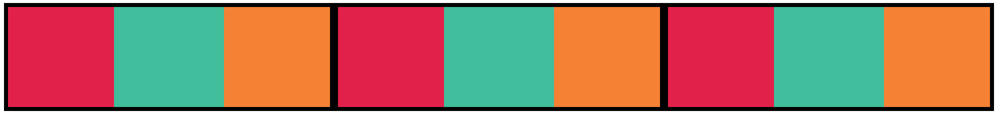


Memory Layout

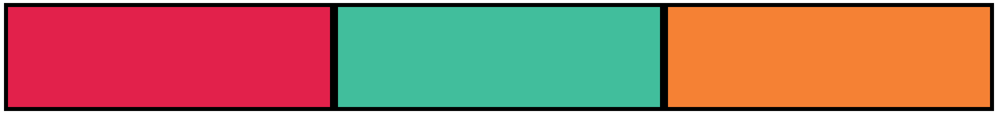
```
struct Cell {
  double a;
  double b;
  double c;
};
```



```
Cell cells[3];
```

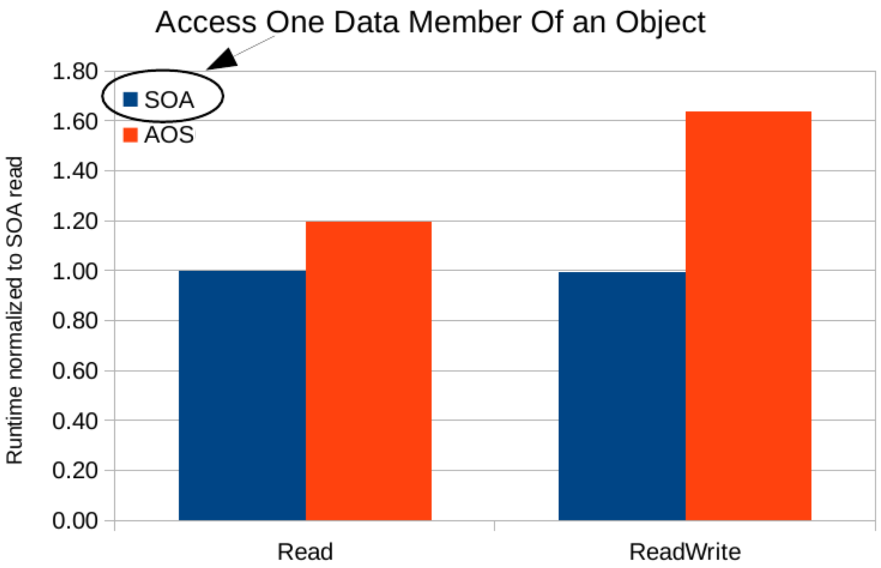


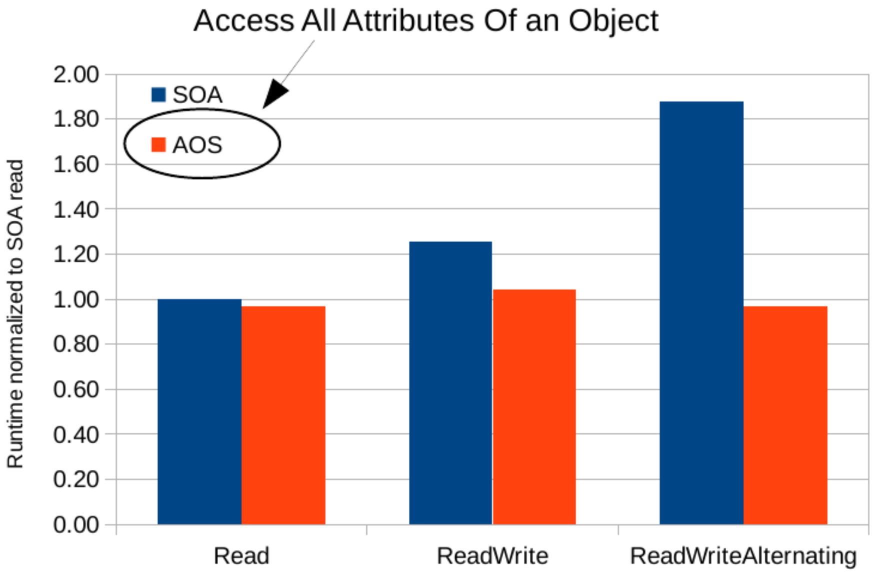
```
struct SoaCell {
  double a[3];
  double b[3];
  double c[3];
};
```



Best memory layout depends on access pattern

- SOA vs. AOS - synthetic benchmark





Preliminary Performance Results

- Comparison with Cortex3D
Speed-up between **7x** and **130x** on one CPU core depending on operation

Number of Cells	Runtime for one iteration	Main memory consumption
2M	150 ms	700 MB
134 M	~9s	~36 GB

Cloud / Distributed Runtime

+ Democratizing HPC access

+ Elasticity

+ Scales by credit card

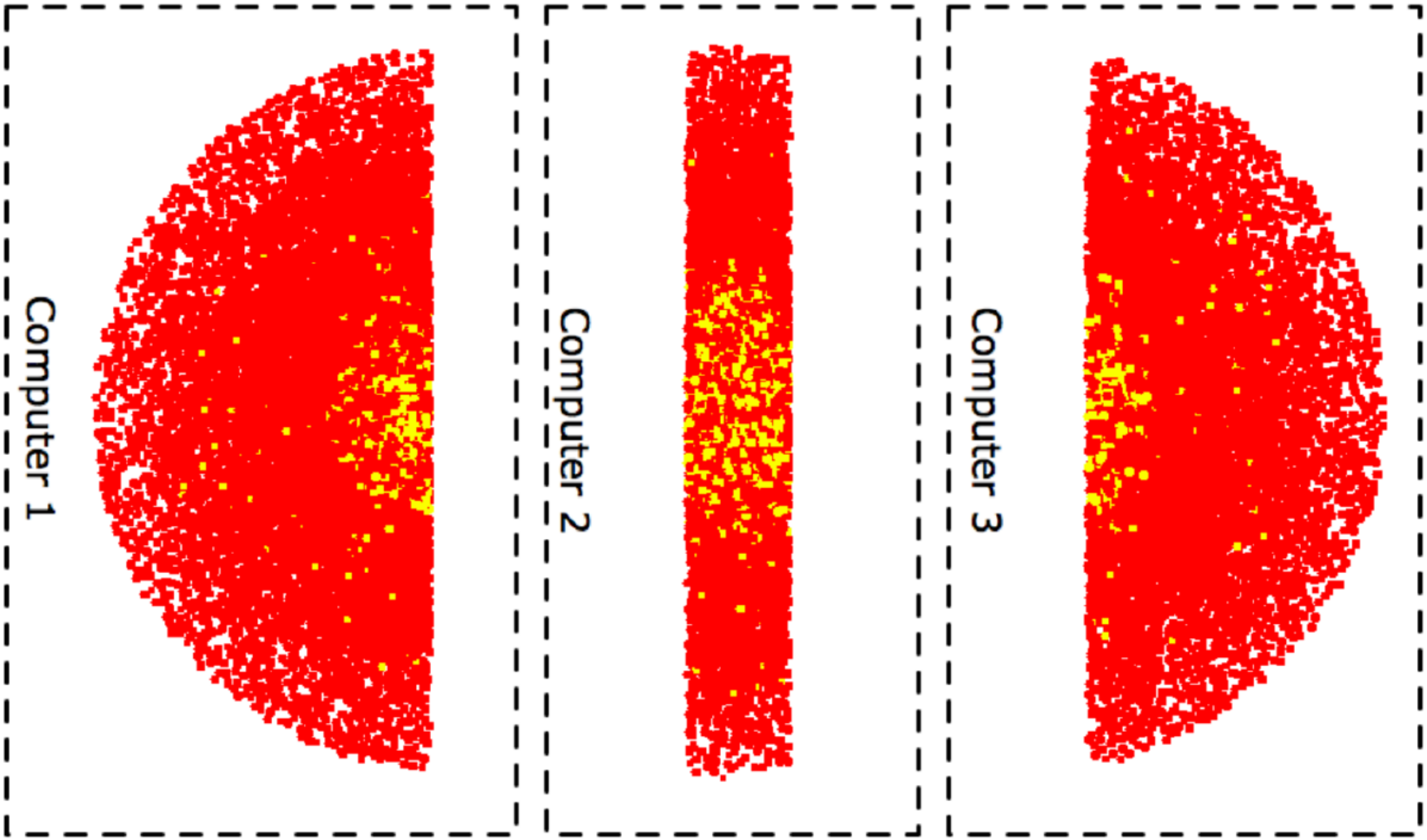
+ Reduced IT administration
  (support, technology upgrades)

- Failure rate

- low bandwith

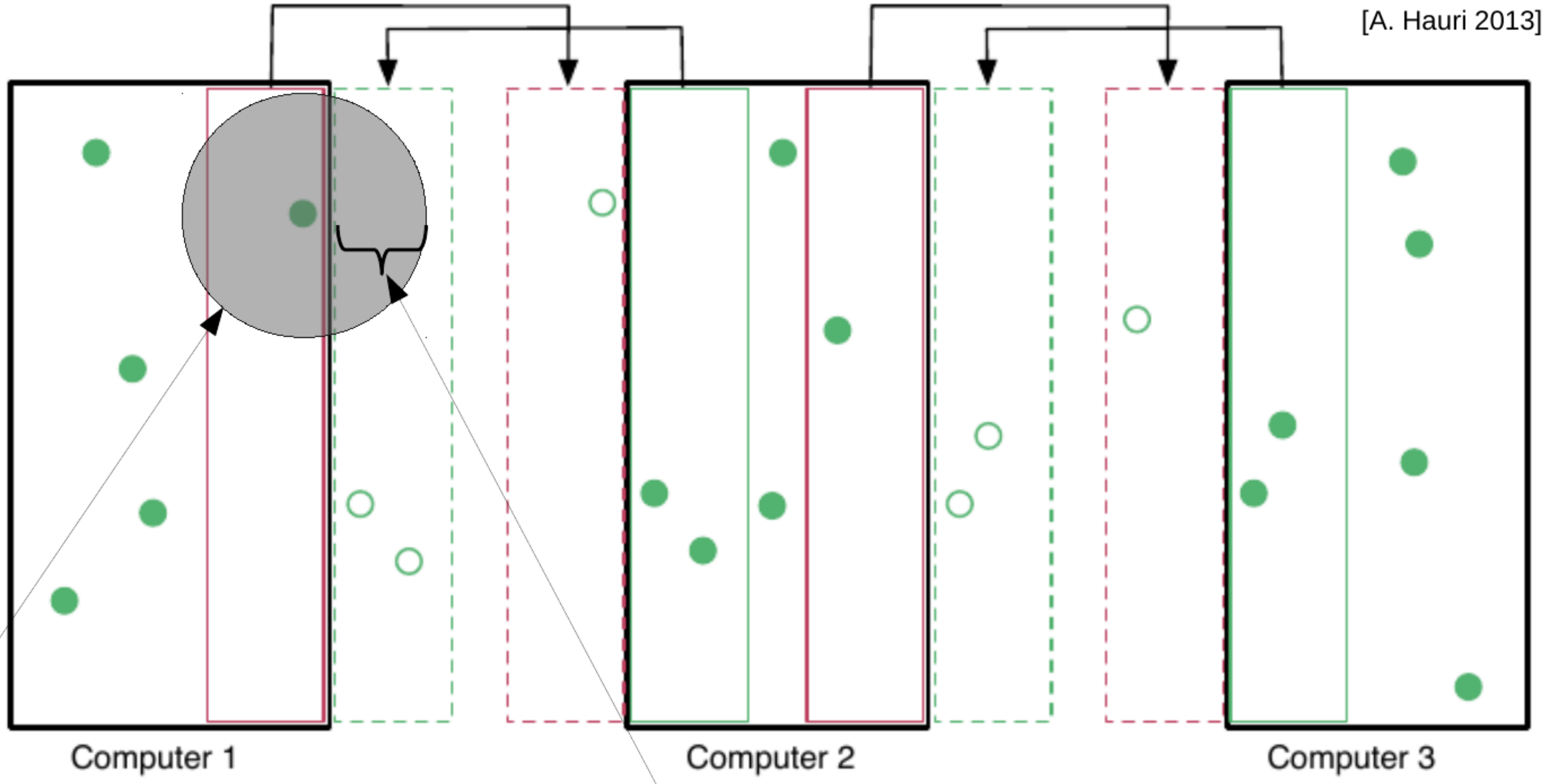
- high latency

Space Partitioning



Hauri, Andreas. *Self-construction in the context of cortical growth*. Diss. 2013.

Data in the border region must be exchanged after **each** interation

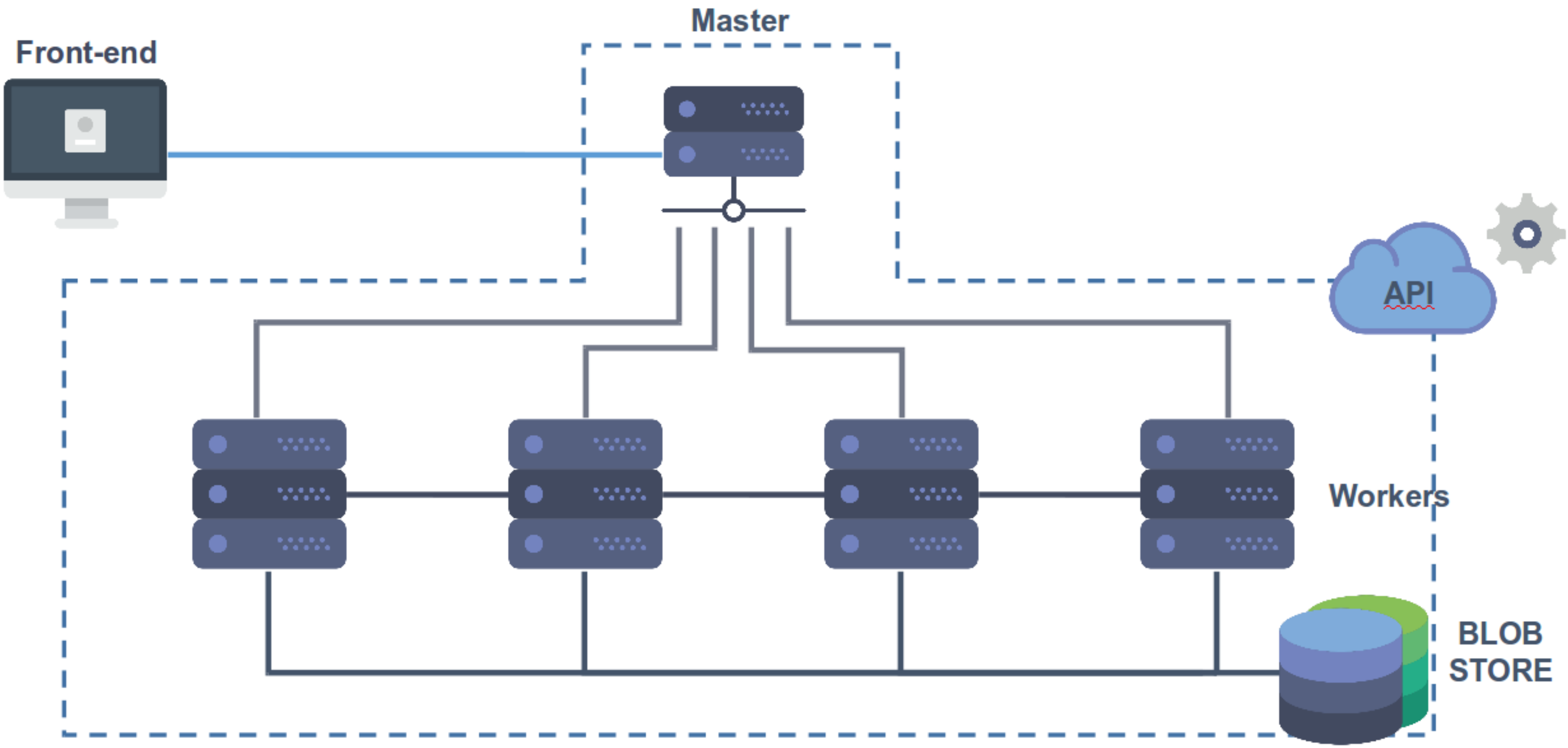


[A. Hauri 2013]

Data required to calculate update of this cell

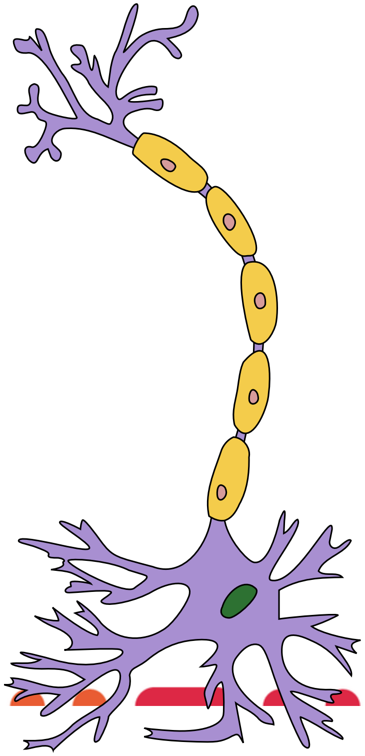
Data originally not available on computer 1 needs to be transferred from computer 2

Distributed Runtime



Flexibility / Modular Architecture

- Custom **simulation objects**
- Custom **cell behaviour**
- Custom **events**
  - Cell division
  - Neurite branching
- Custom **process**
  - Diffusion

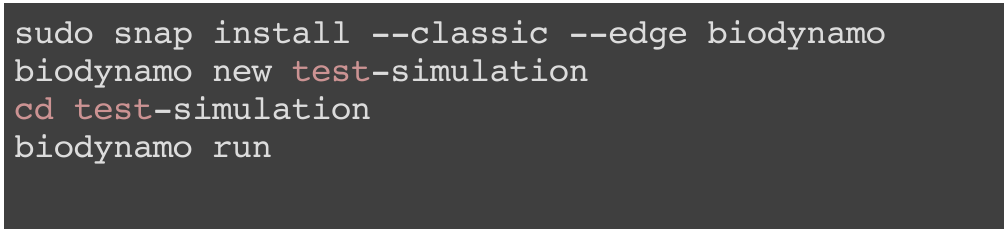


User-friendliness

- Easy installation
- Helper application
- Hidden parallelism
- Predefined functionality / modules

Installation, Compilation and Execution in 4 steps

Linux



```
sudo snap install --classic --edge biodynamo
biodynamo new test-simulation
cd test-simulation
biodynamo run
```

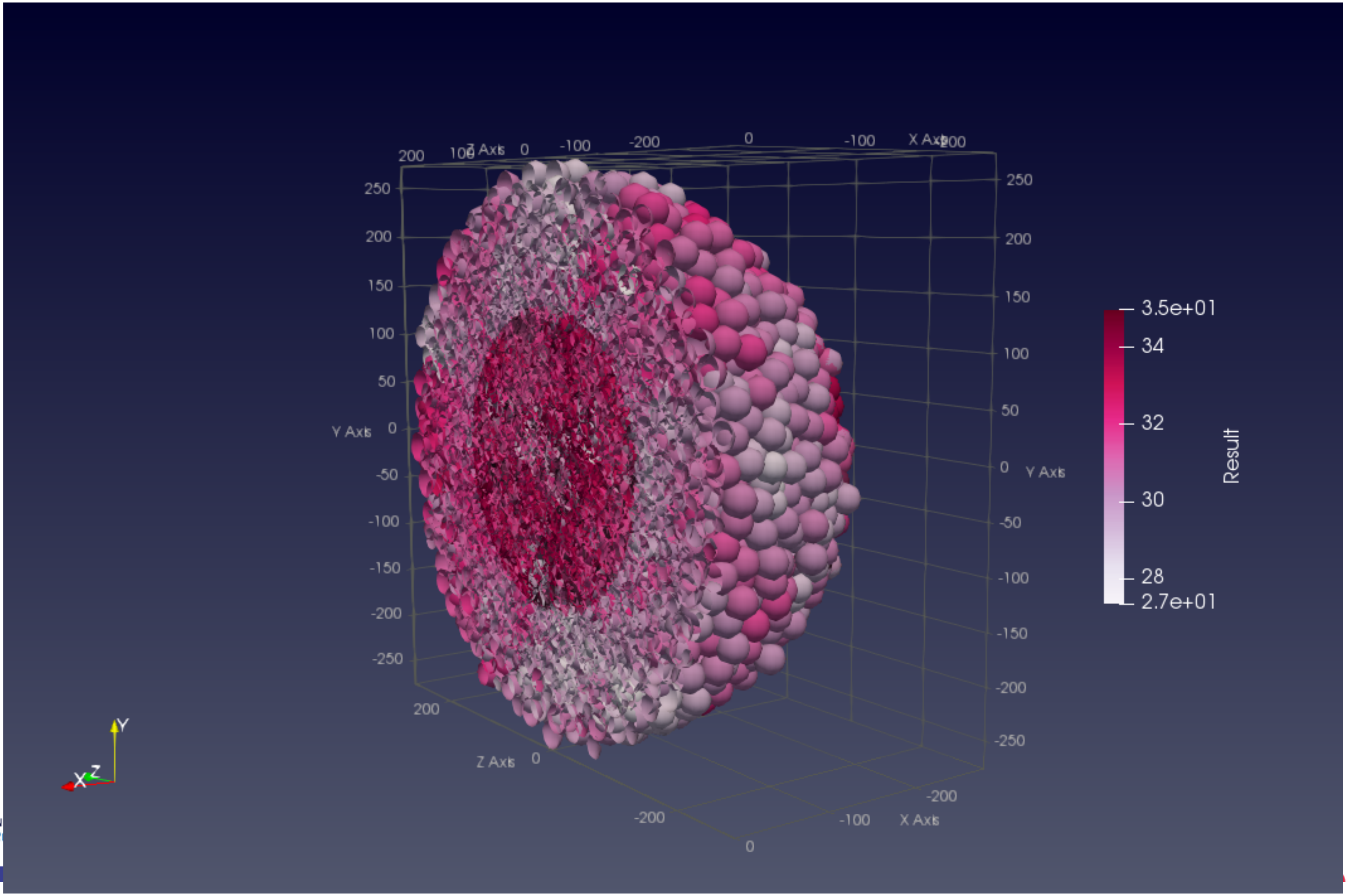
MacOS supported as well

Hello World Example

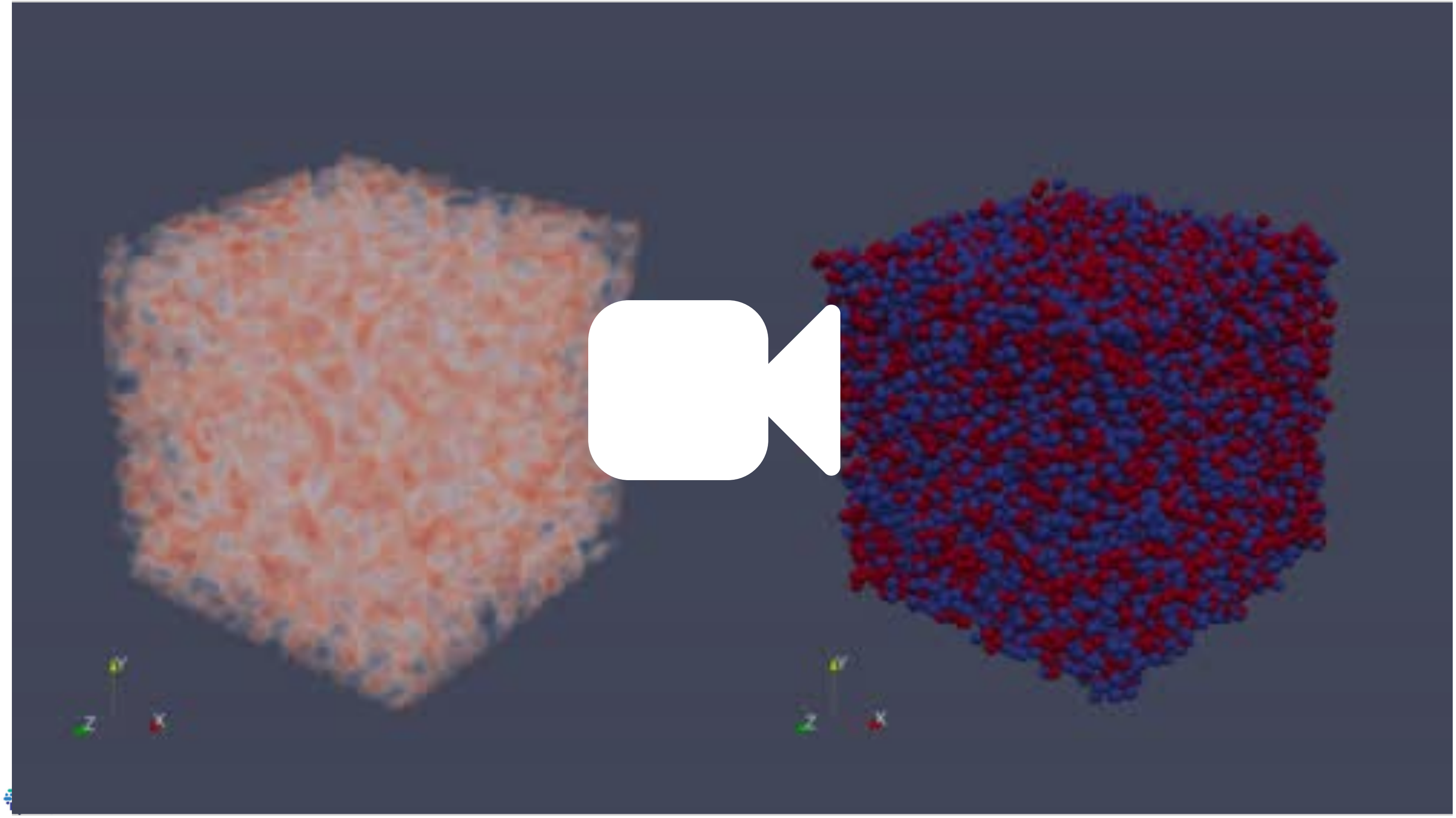
```
...
// Define initial model - in this example: single cell at origin
auto cell = Rm()->New<Cell>({0, 0, 0});
cell.SetDiameter(30);
cell.AddBiologyModule(GrowDivide());

// Run simulation for one timestep
Scheduler<> scheduler;
scheduler.Simulate(1);
...
```

Visualization



Visualization



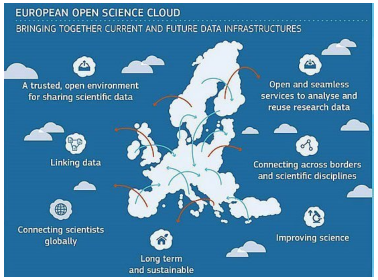
Overview Current Features

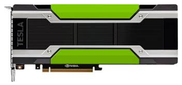
- **Modular simulation engine**
  - User defined cell behavior, cell types…
  - Mechanical interactions
  - Extracellular diffusion
- ROOT I/O for simulation **backups** , …
- **Fully parallelized** with OpenMP
- **Visualization** using ParaView
- Prototype of **messaging layer** using ZeroMQ

Still early stage

Future Challenges / Vision

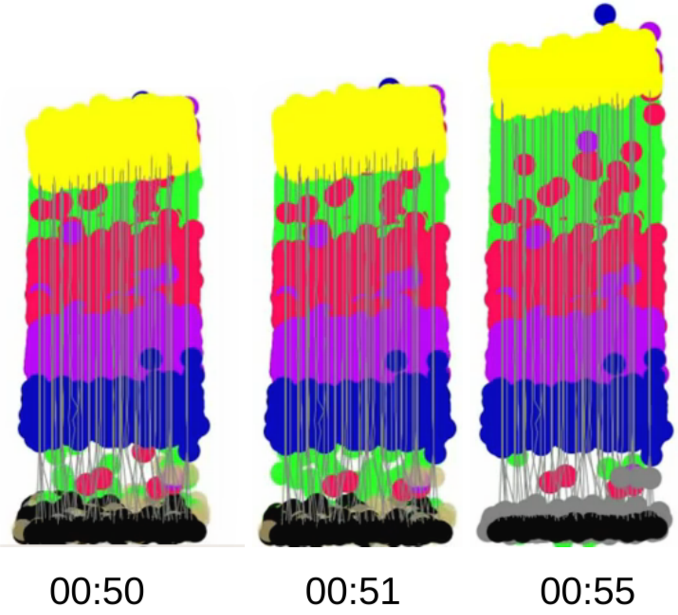
- Cloud computing
- Heterogeneous computing
- Interfacing with well-established knowledge bases


EUROPEAN OPEN SCIENCE CLOUD
BRINGING TOGETHER CURRENT AND FUTURE DATA INFRASTRUCTURES
A trusted, open environment for sharing scientific data
Open and seamless services to analyse and reuse research data
Linking data
Connecting across borders and scientific disciplines
Connecting scientists globally
Improving science
Long term and sustainable



Future Challenge: Cloud Computing

- **Expectation**
  Network IO will be bottleneck for simulations
- **Hypothesis**
  Minimize data transfer based on the idea that:
  - not all attributes will change during one iteration
  - changes are incremental
  - might be predicted



https://www.youtube.com/watch?v=9InvFfnAkus

Future Challenge: Heterogeneous Computing

Computing hardware:

- CPU
- GPU
- FPGA
- DSP

Physics on GPU: promising preliminary results!

Summary

- High-performance simulation platform
- Modular architecture: more efficient workflow
- Platform in begin phase, but already usable
- Distributed runtime in near future
- Heterogeneous computing is upcoming

QUESTIONS?

lukas.breitwieser@cern.ch

ahmad.hesam@cern.ch