Computational steering: Interactive Design-through-Analysis

Casper van Leeuwen (SURF) and Matthias Möller (TU Delft)

SURF

TU Delft

SURF Open Innovation Lab

Explore technological innovations for education and research with focus on the challenges of the future

Innovation themes

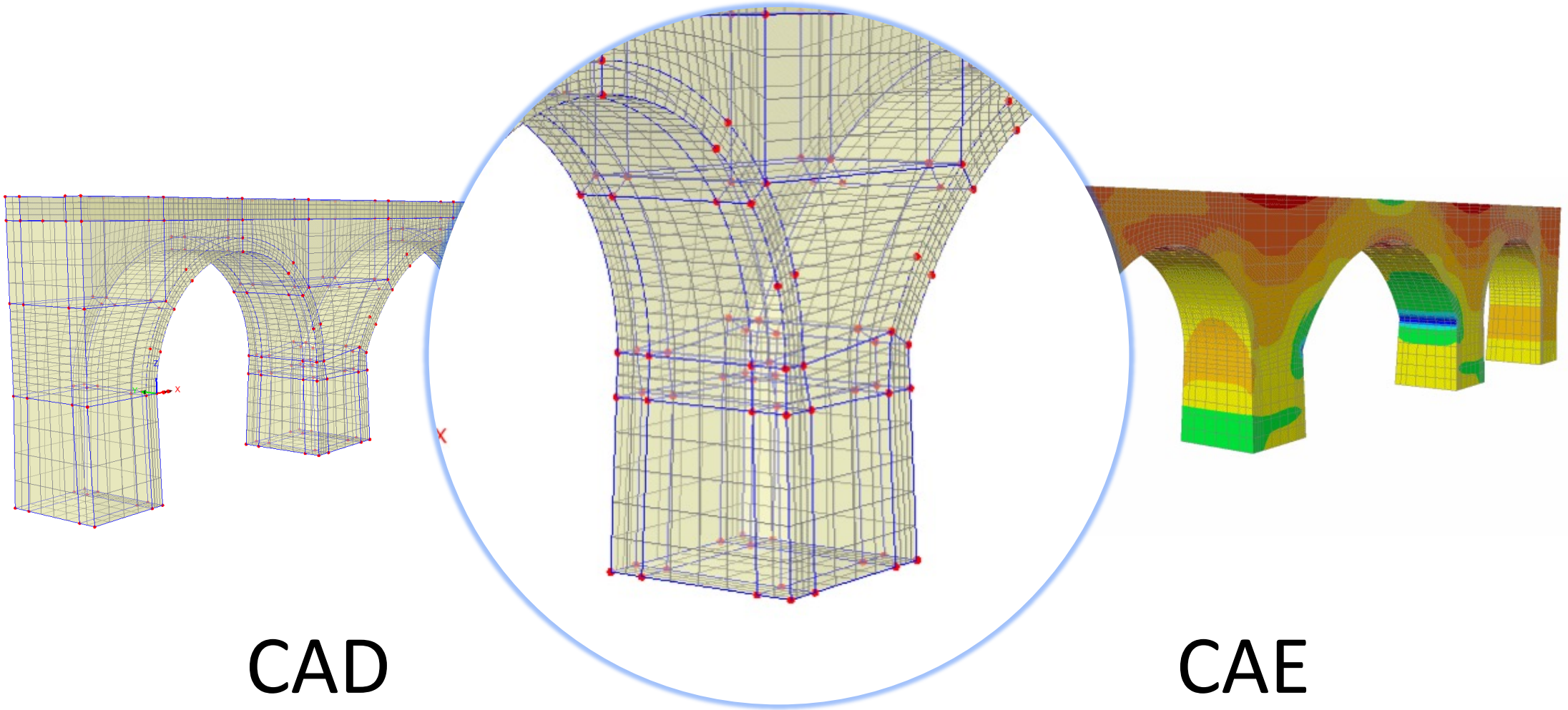
- Future computing & networking
- Edge-oriented technologies
- Data processing & data management technologies
- Artificial Intelligence


Design a
bottle


SUMMER
DRAFT


Design a
bridge

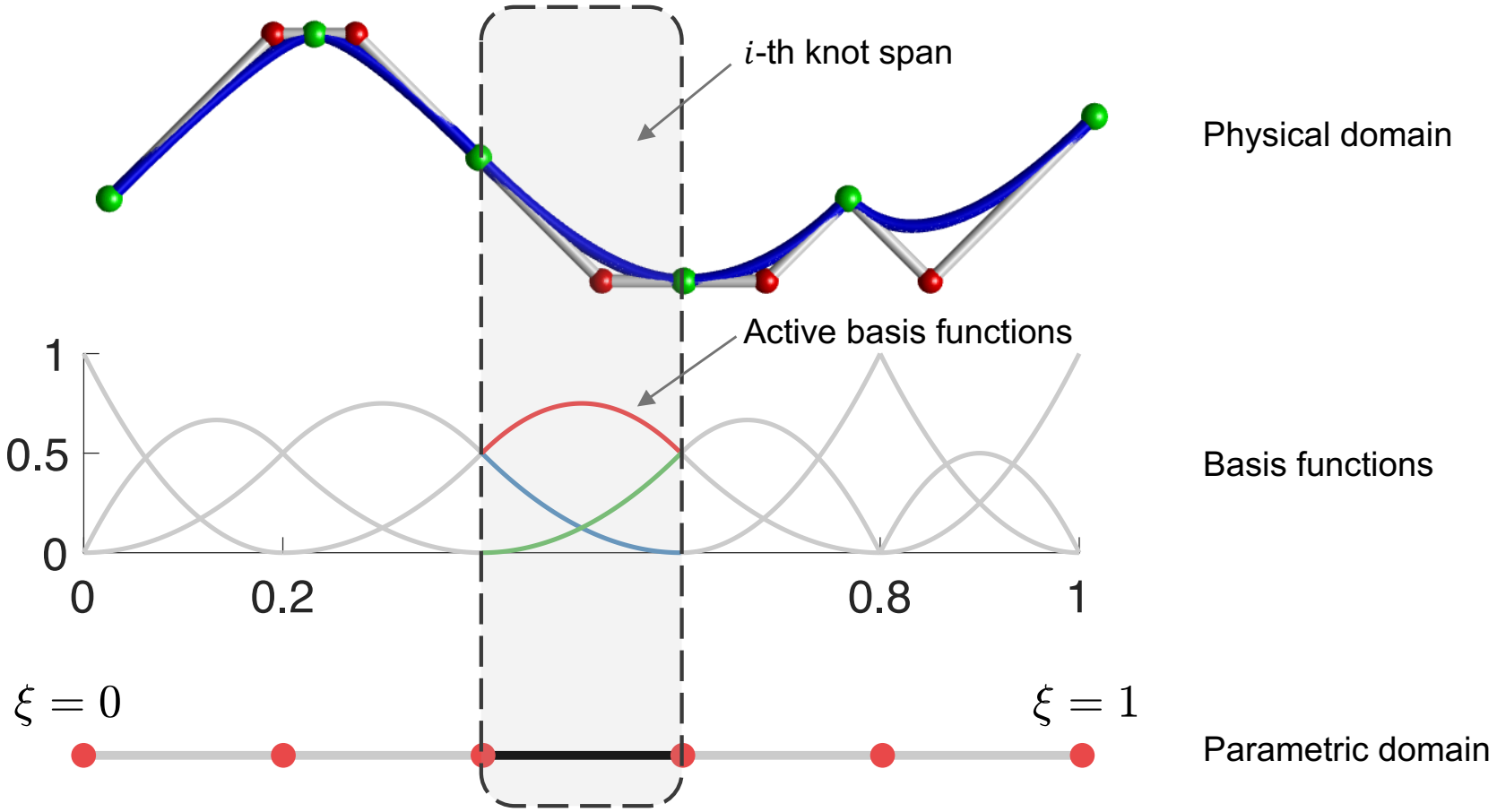
Design-through-analysis



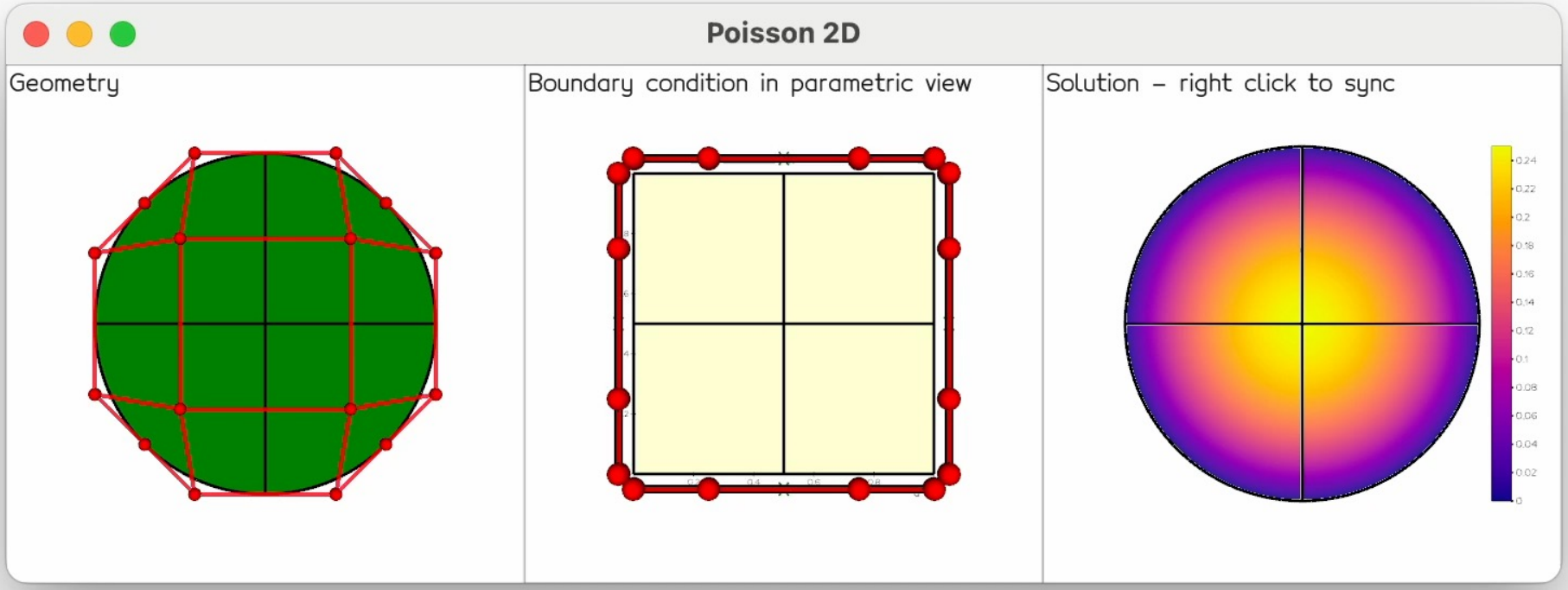
CAD

CAE

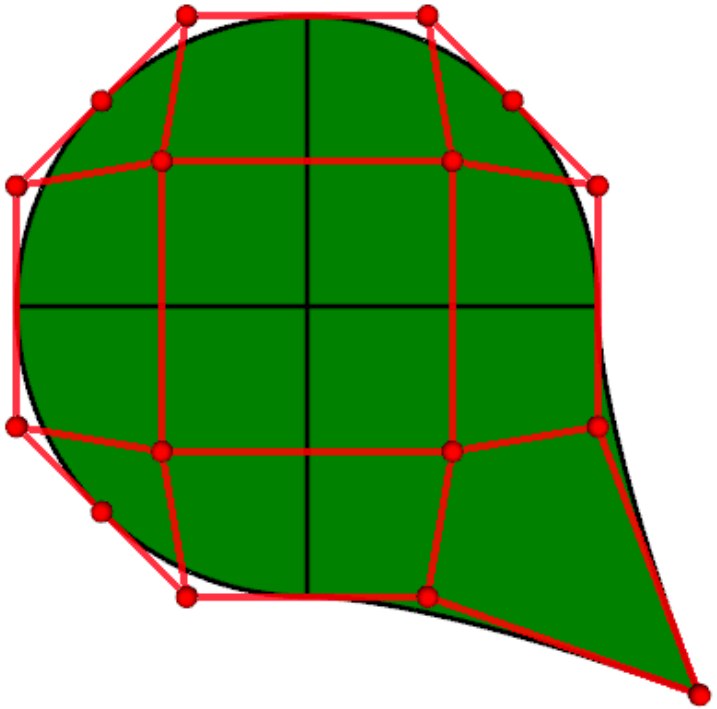
Spline curves $C(\xi) = \sum_{i=1}^{n} \boldsymbol{P_i} b_i(\xi)$



CAD + CAE with splines



The math behind it

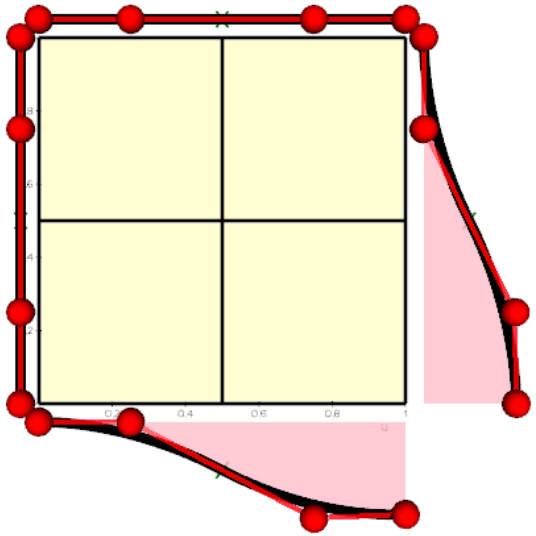


$$\boldsymbol{x}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{x}_{i,j} b_i(\xi) b_j(\eta)$$

The math behind it

$$u(\xi, 1) = \sum_{i=1}^{n} g_{i,m} b_i(\xi)$$

$$u(0, \eta) = \sum_{j=1}^{m} g_{1,j} b_j(\eta)$$



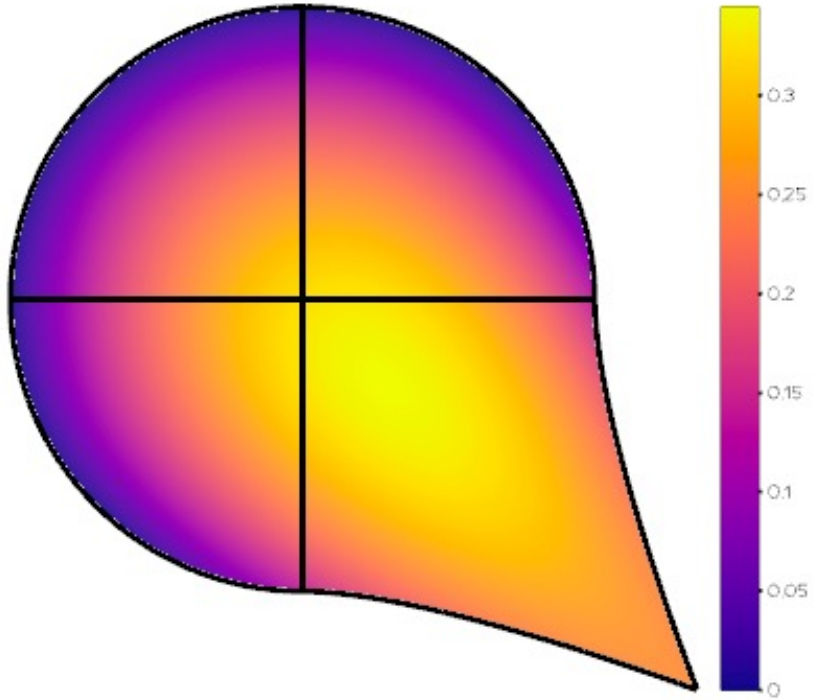
$$u(1, \eta) = \sum_{j=1}^{m} g_{n,j} b_j(\eta)$$

$$u(\xi, 0) = \sum_{i=1}^{n} g_{i,1} b_i(\xi)$$

The math behind it

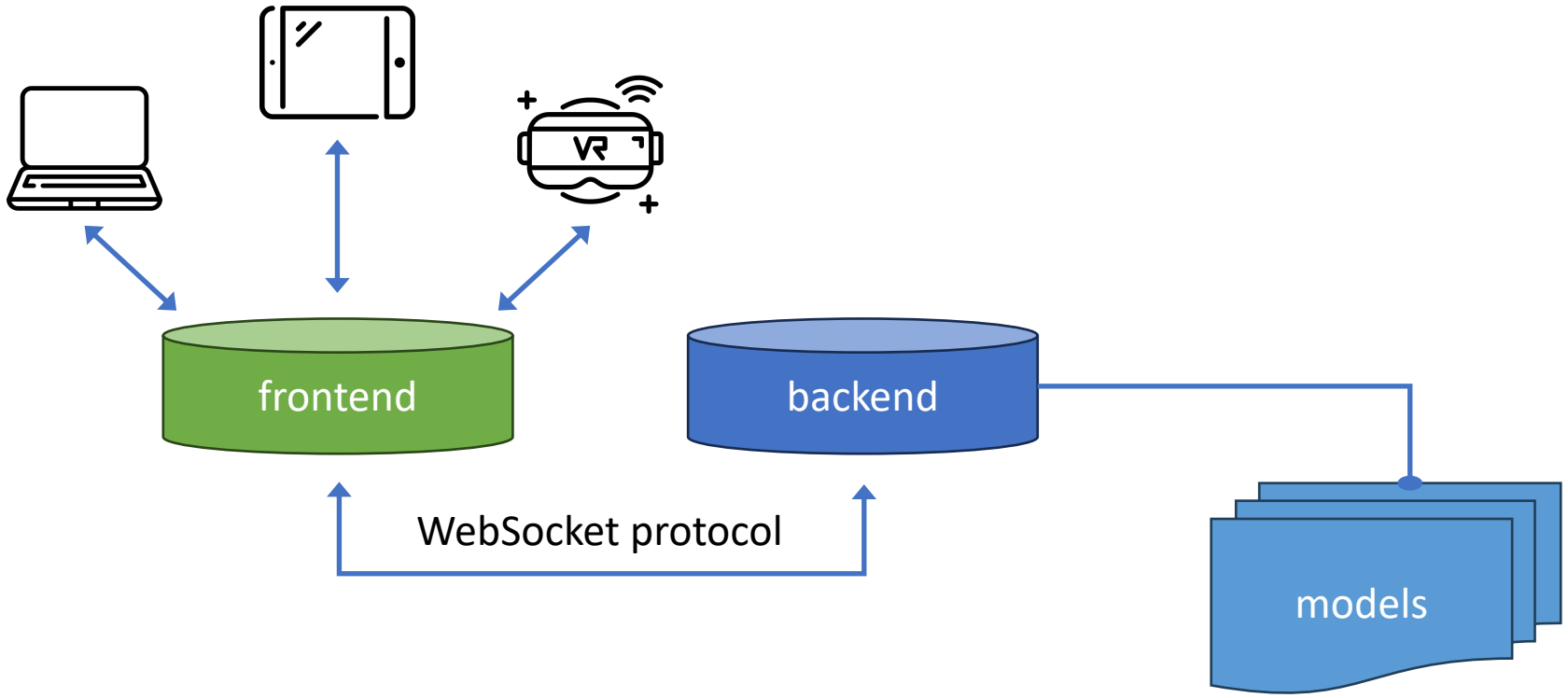
$$-\Delta u = f \text{ in } \Omega$$
$$u = g \text{ on } \Gamma$$

$Ax = b$

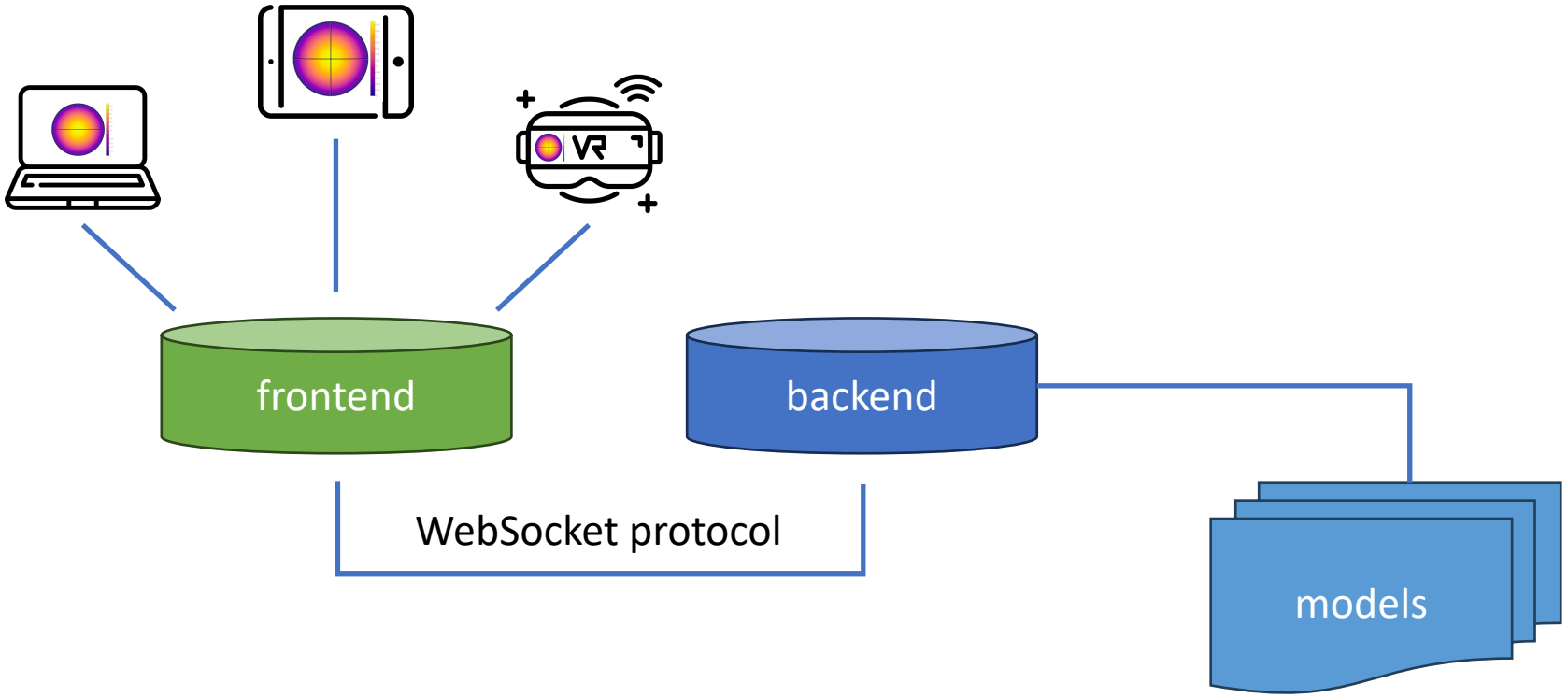


$$u(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} u_{i,j} b_i(\xi) b_j(\eta)$$

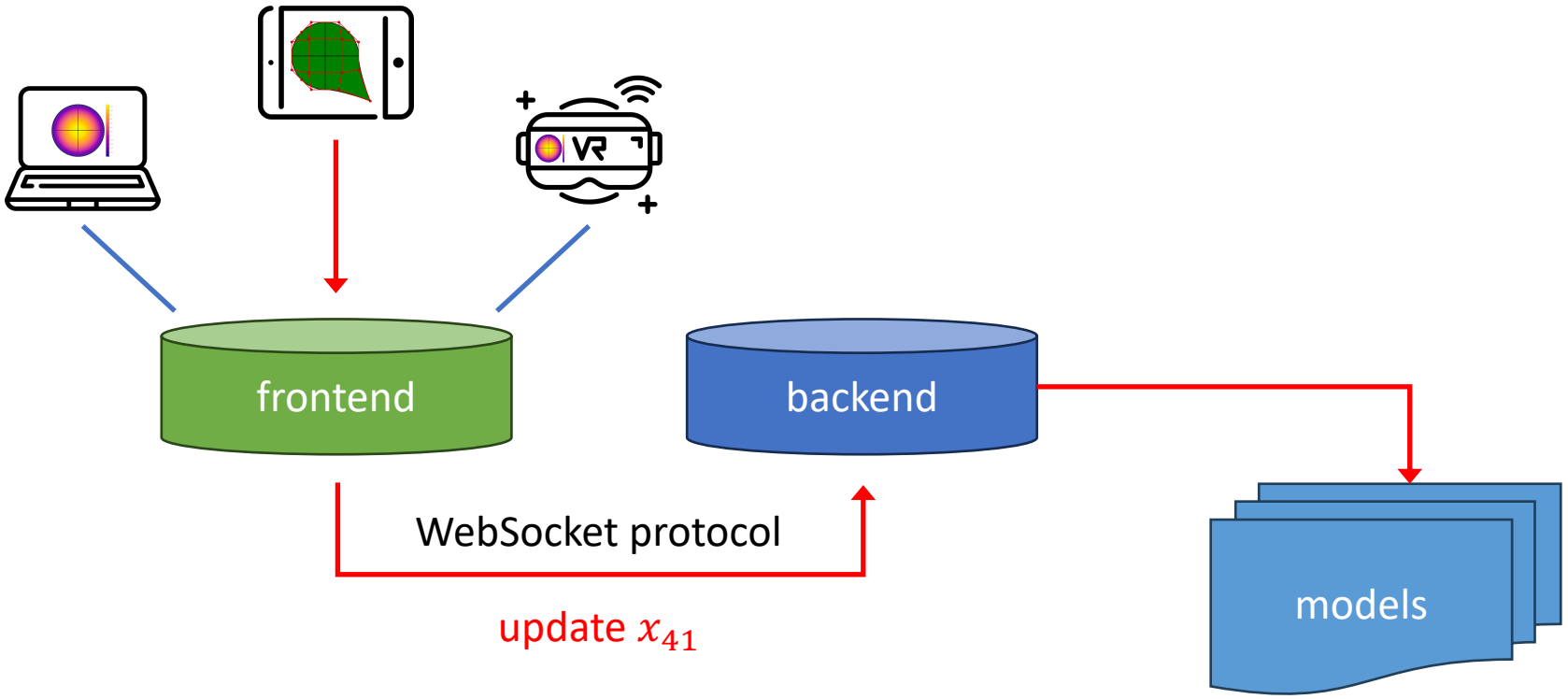
Collaborative DTA workflow



Collaborative DTA workflow

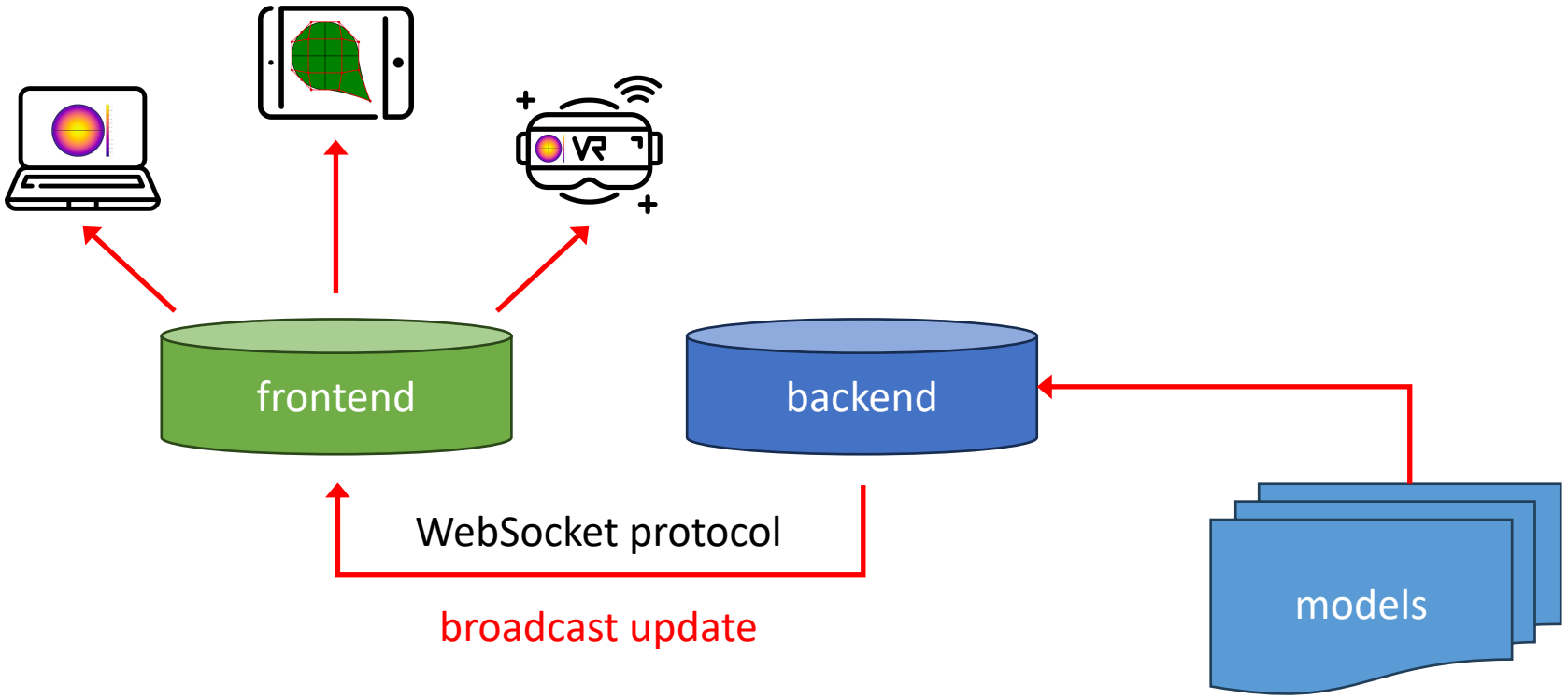


Collaborative DTA workflow

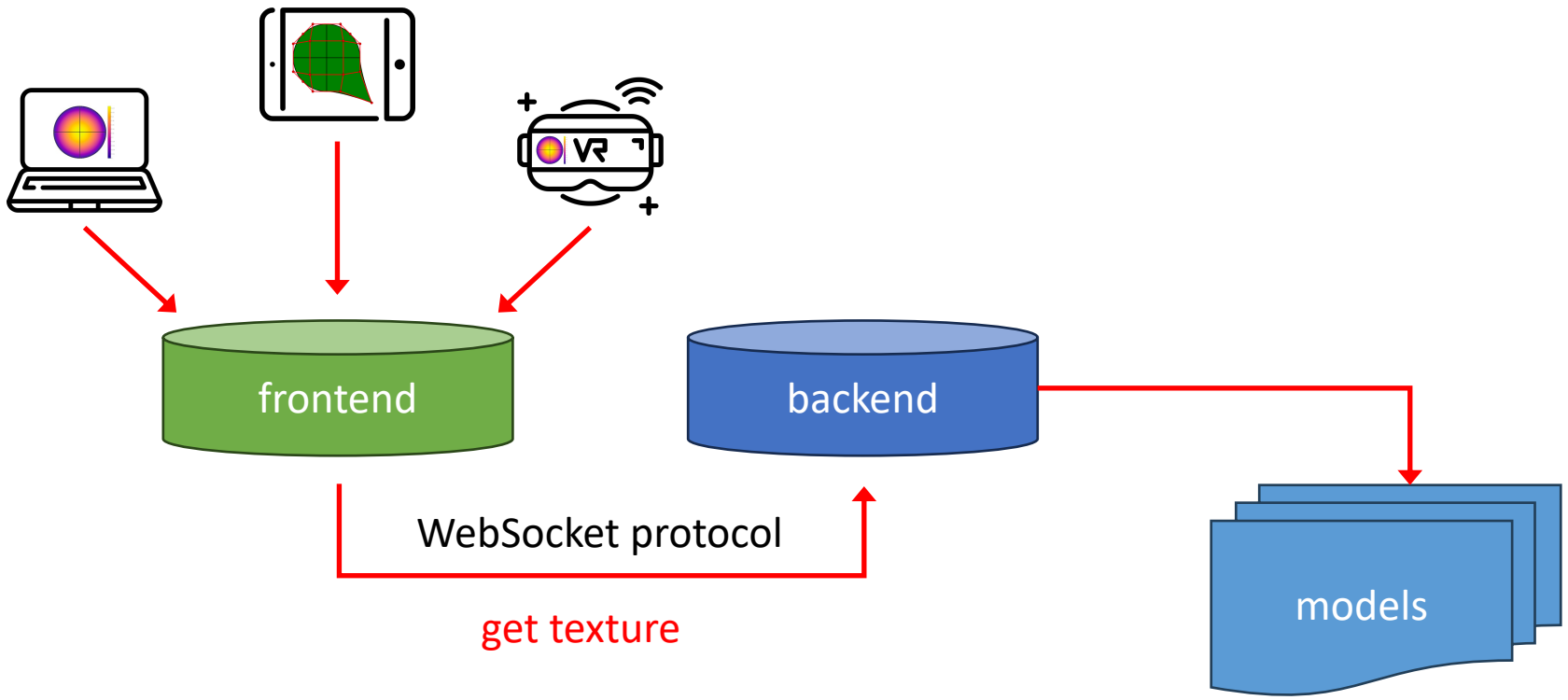


- update geometry
- update solution

Collaborative DTA workflow

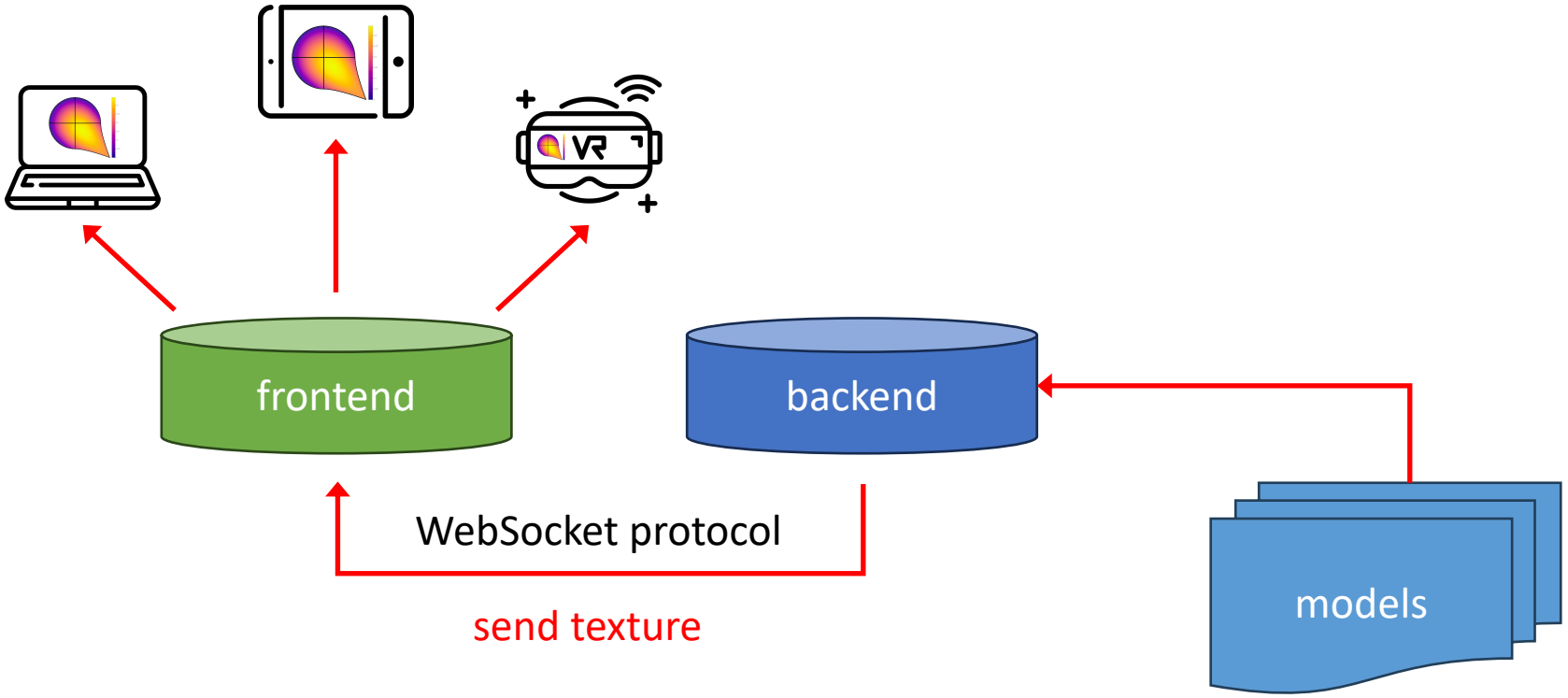


Collaborative DTA workflow



- create texture

Collaborative DTA workflow



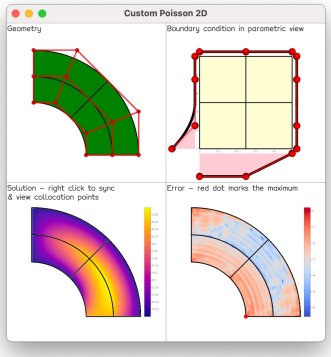
Collaborative DTA workflow

C++17 threads
μWebSockets

frontend

backend

Python



WebSocket protocol

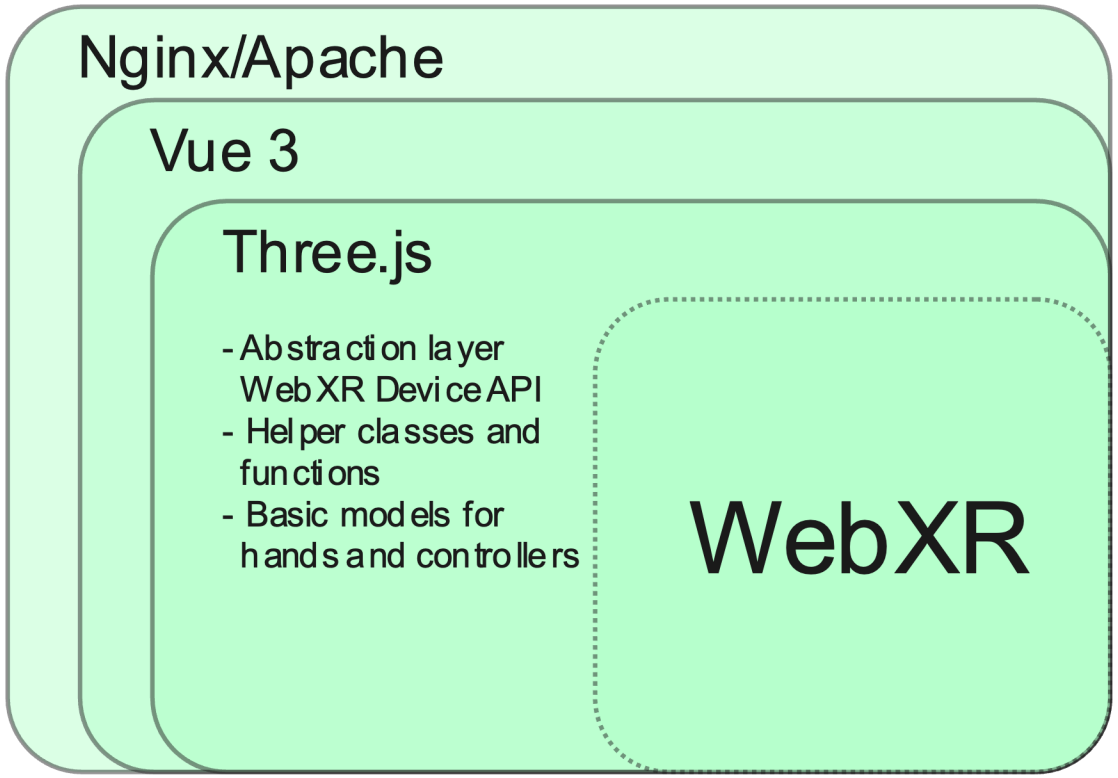
models

Dynamic libraries

IgANet

WebXR implementation

Our implementation:



Possible other solutions:

- Babylon.js (low-code editor)
- PlayCanvas (free public, paid private low-code editor)
- A-frame (HTML-document style)

Live demo