FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Terra-Neo - Integrated Co-Design of an Exascale Earth Mantle Modeling Framework

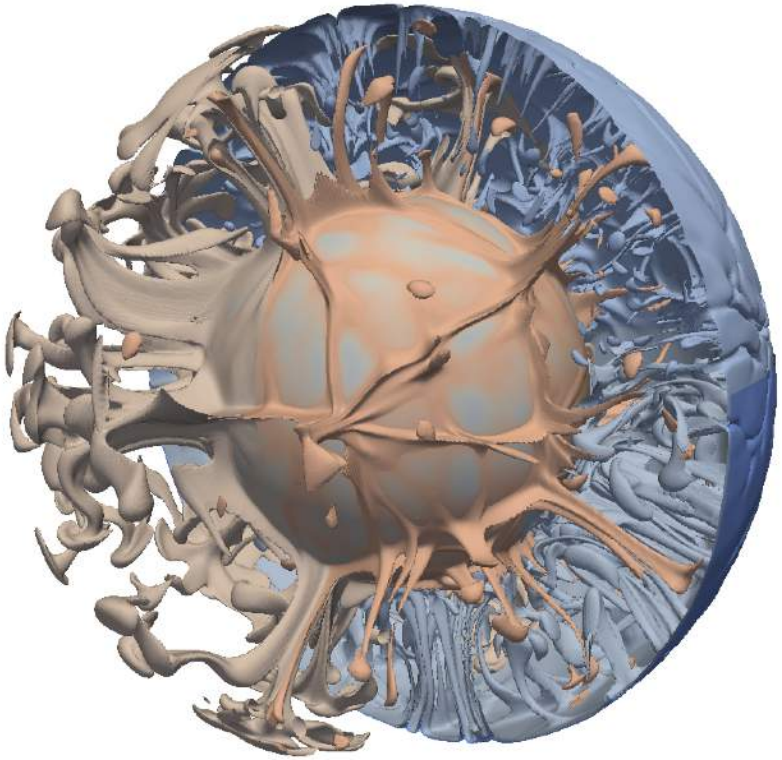
Nils Kohl, D. Thönnes, D. Drzisga (TUM), D. Bartuschat, U. Rüde
Chair for System Simulation, University of Erlangen-Nürnberg
March 21, 2018

SPPEXA Annual Plenary Meeting 2018
Garching, Germany

joint work with
Markus Huber, Barbara Wohlmuth (TUM)
Simon Bauer, Marcus Mohr, Hans-Peter Bunge (LMU)

TerraNeo Project

Motivation: simulation of earth mantle convection



[Source: Terra-Neo reference implementation]

TerraNeo Project

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

re-design?

Problems with reference code:

- not extensible to higher order FE
- ill-designed load balancing approach
- hard to maintain

Fast progress through reuse of well-maintained waLBerla (http://walberla.net) core library!

TERRANEO

TerraNeo Project

Building Blocks

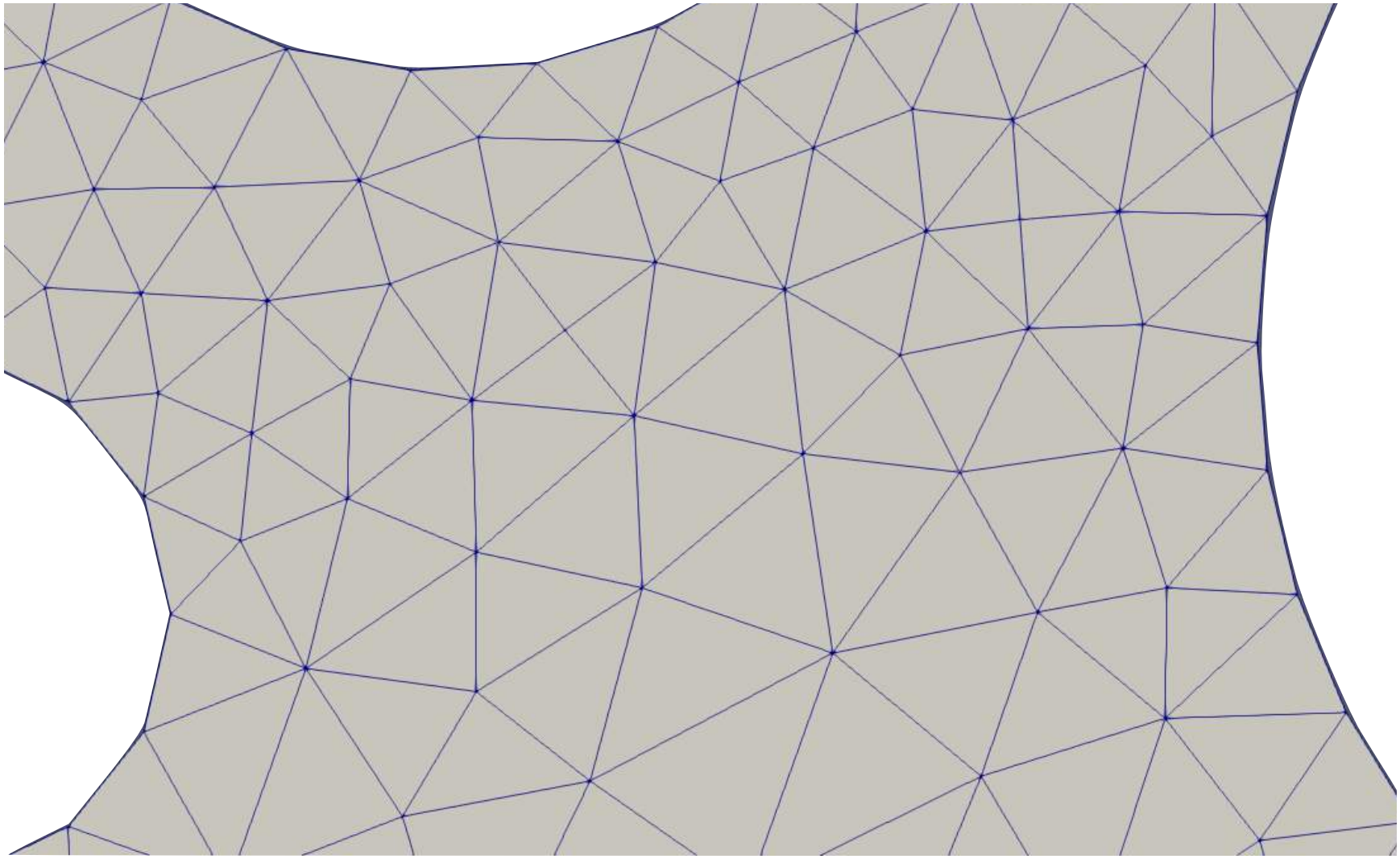
- hierarchical hybrid grids
- fully distributed data structures
- support for different discretizations

TerraNeo Project

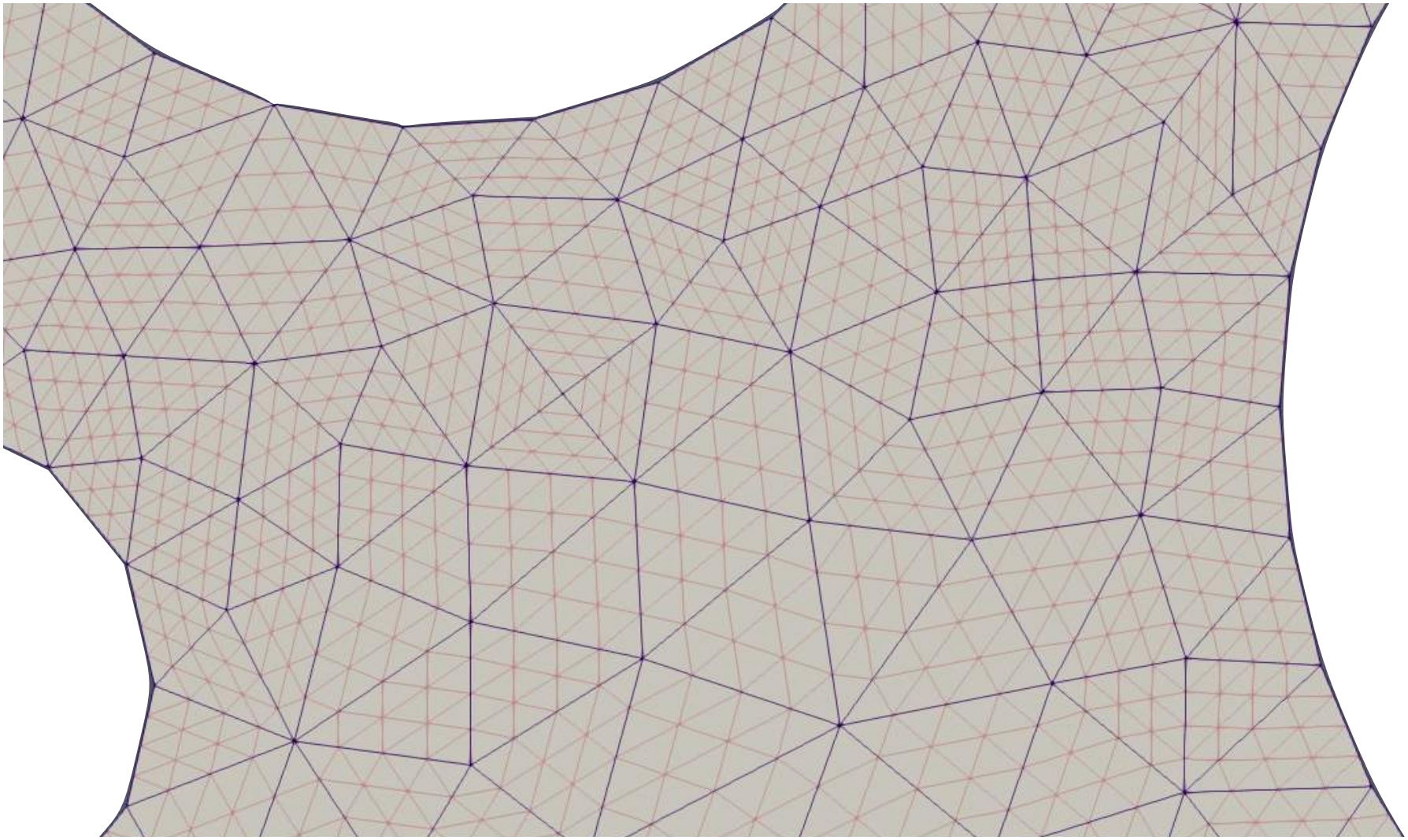
Building Blocks

- hierarchical hybrid grids
- fully distributed data structures
- support for different discretizations

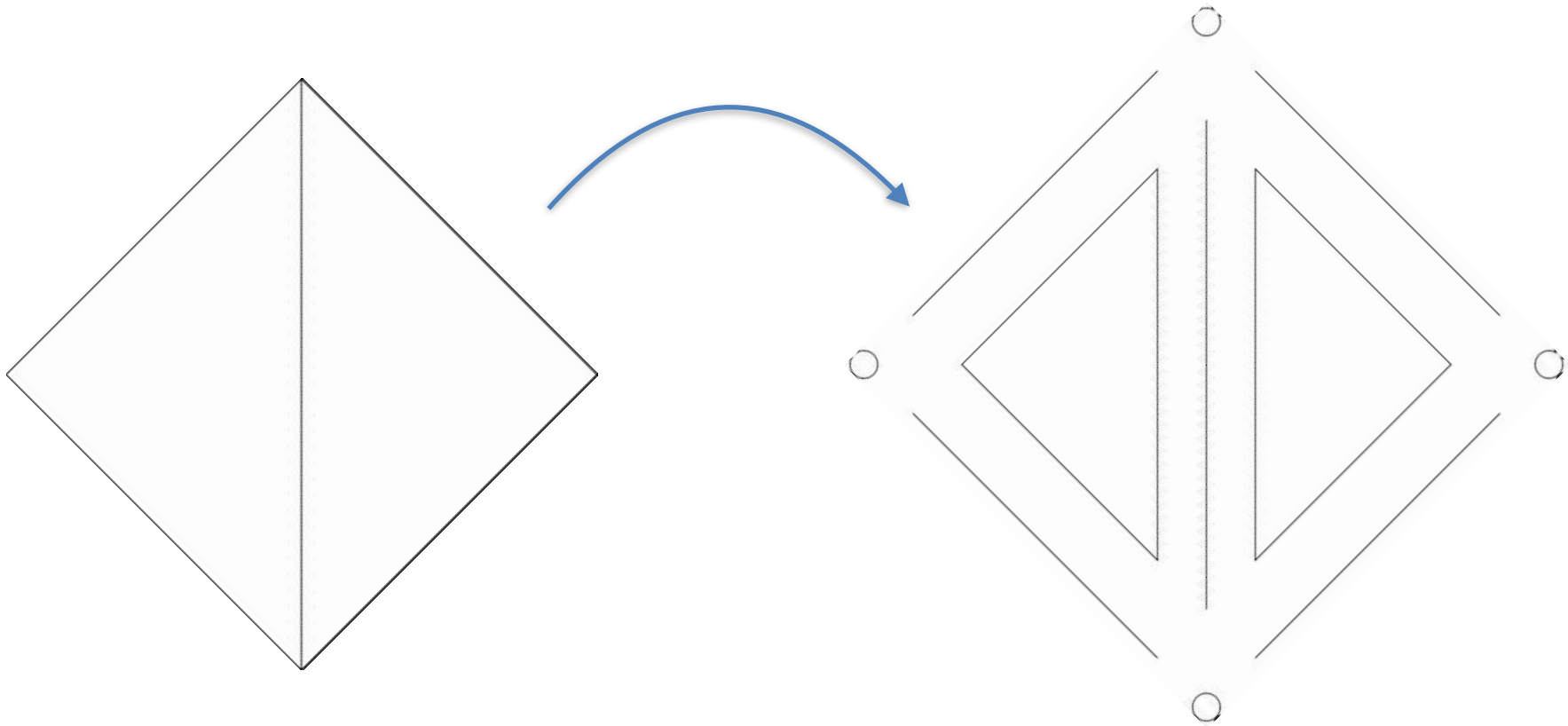
Hierarchical Hybrid Grids



Hierarchical Hybrid Grids



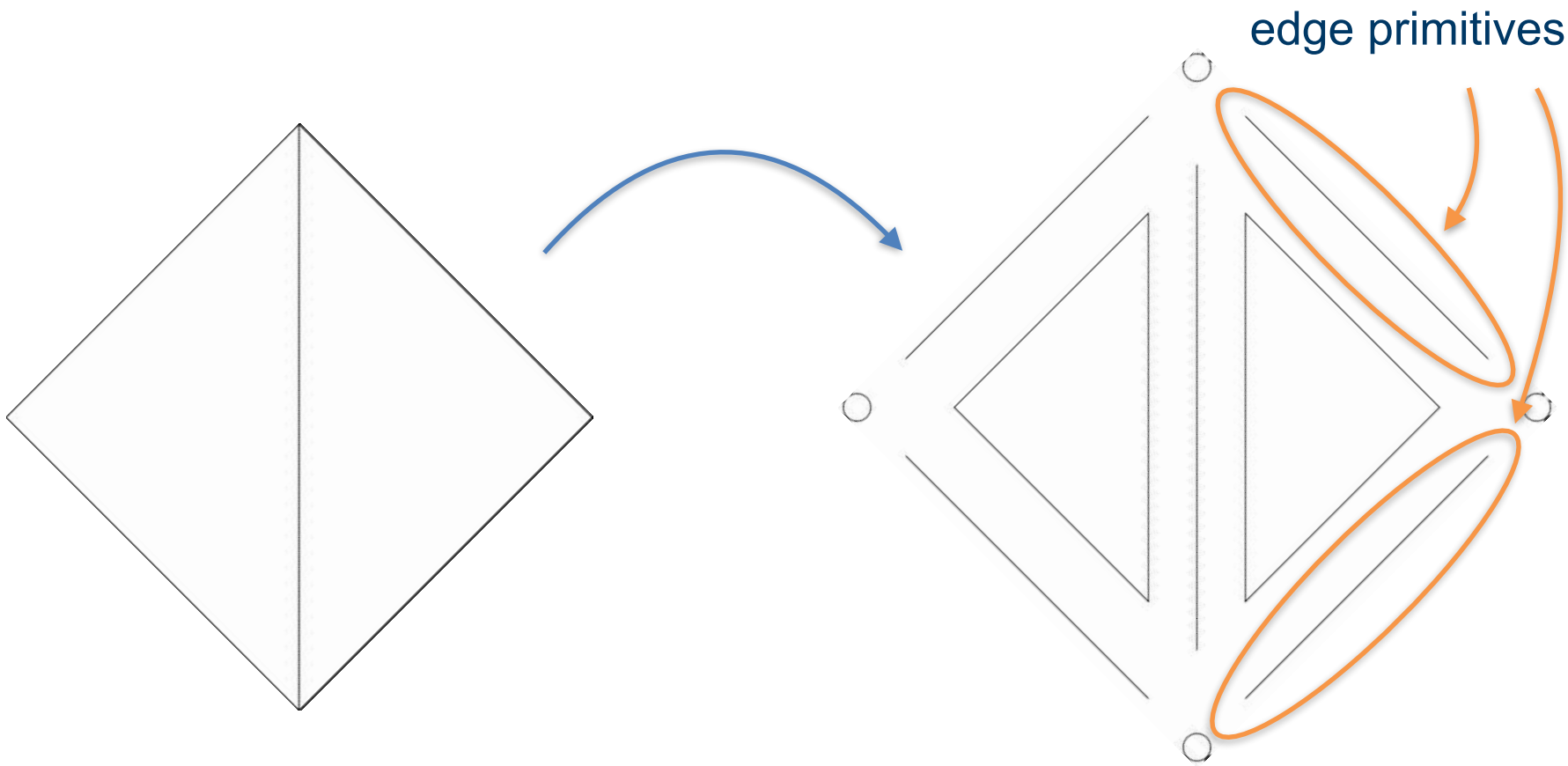
Hierarchical Hybrid Grids



unstructured grid

graph of *primitives*

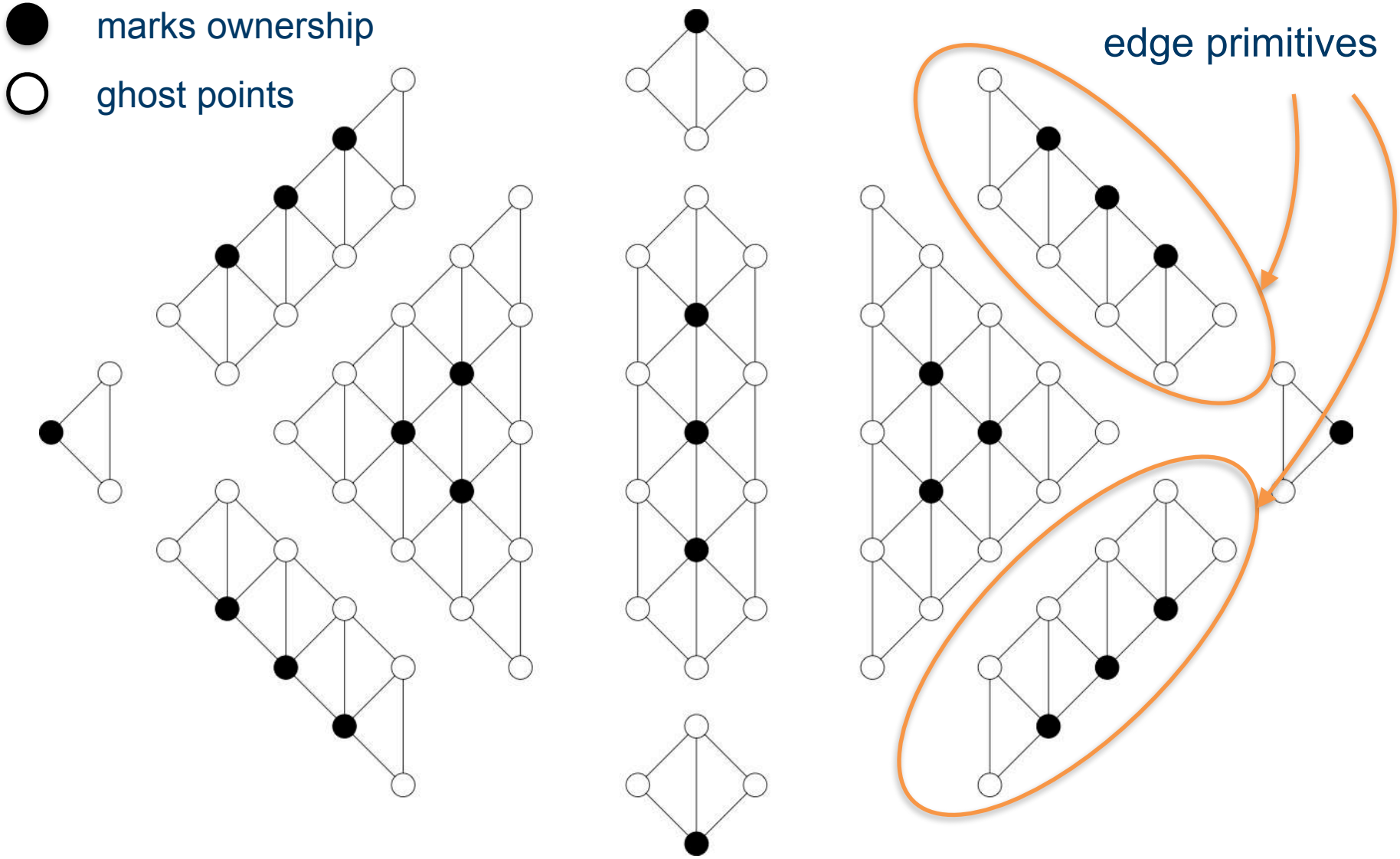
Hierarchical Hybrid Grids



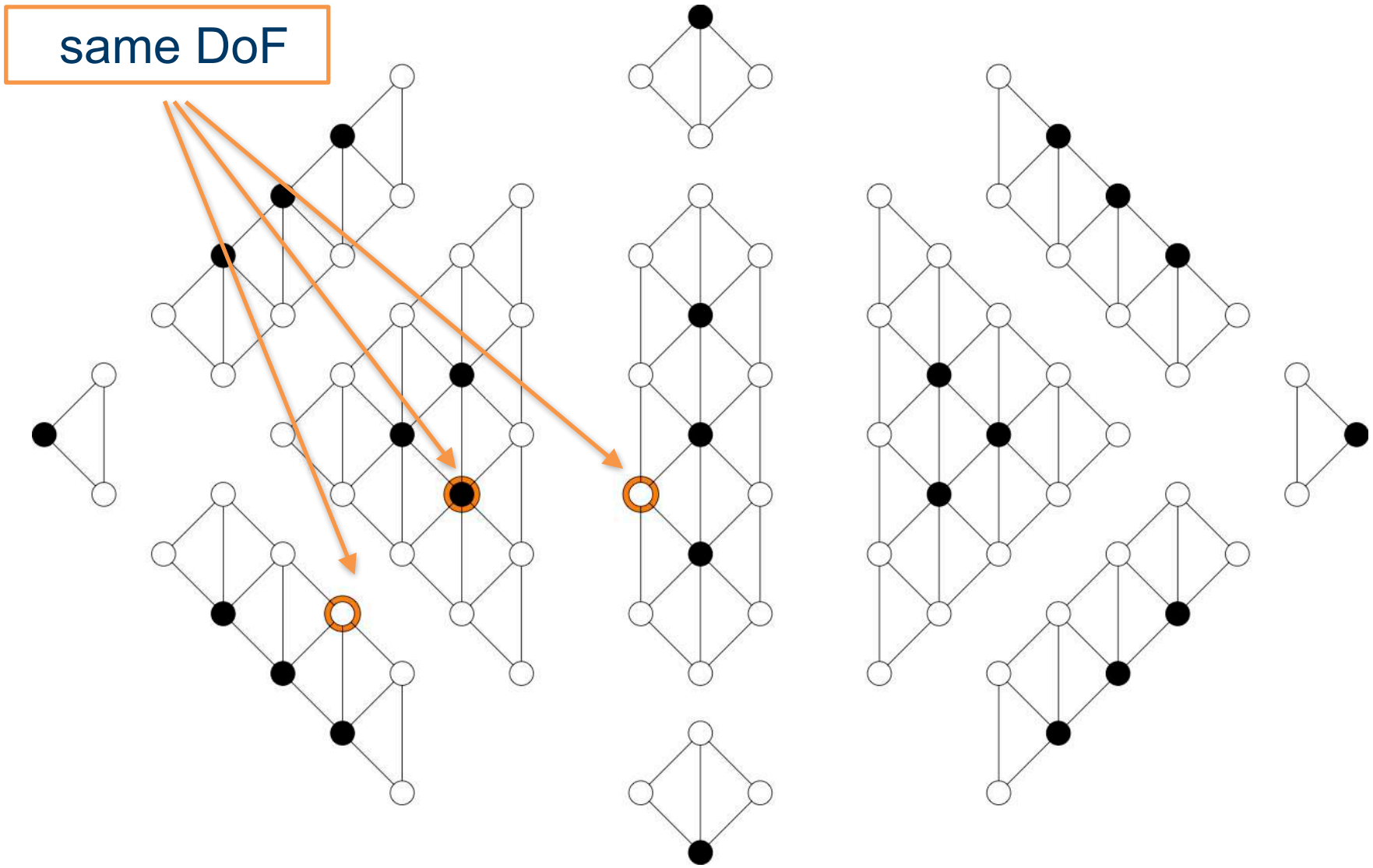
unstructured grid

graph of *primitives*

Hierarchical Hybrid Grids



Hierarchical Hybrid Grids

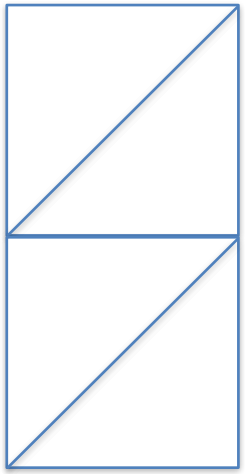


TerraNeo Project

Building Blocks

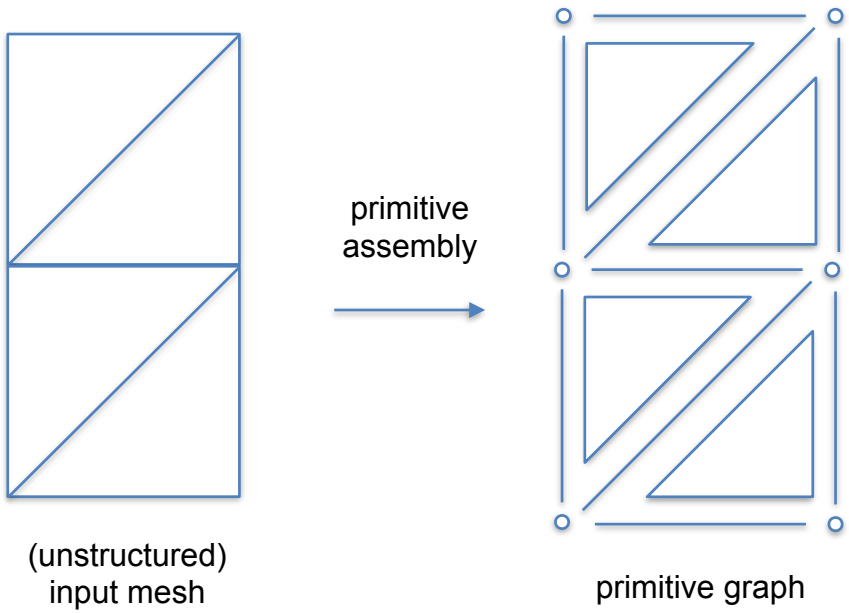
- hierarchical hybrid grids
- fully distributed data structures
- support for different discretizations

From mesh to primitives (2D)

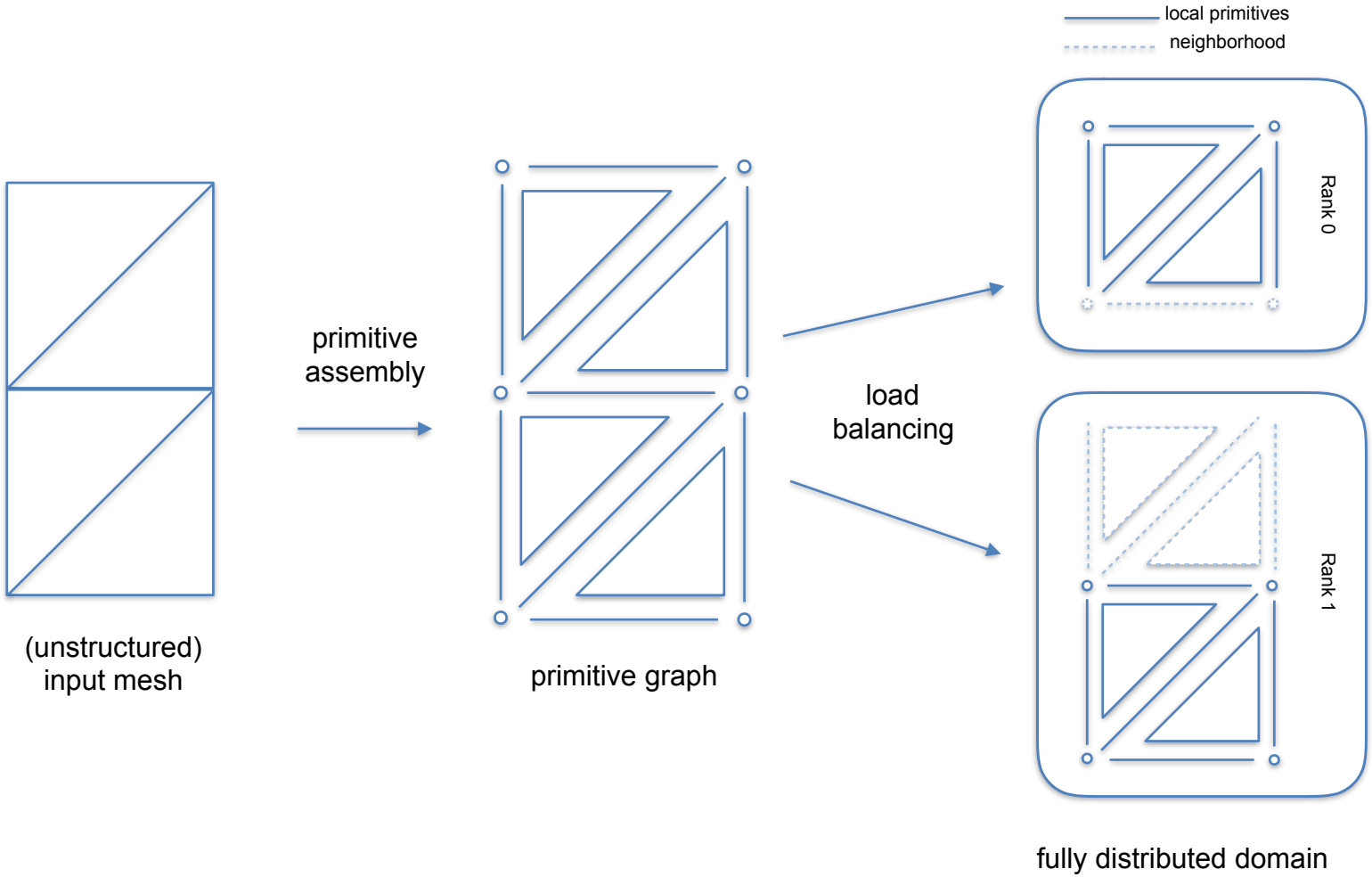


(unstructured)
input mesh

From mesh to primitives (2D)

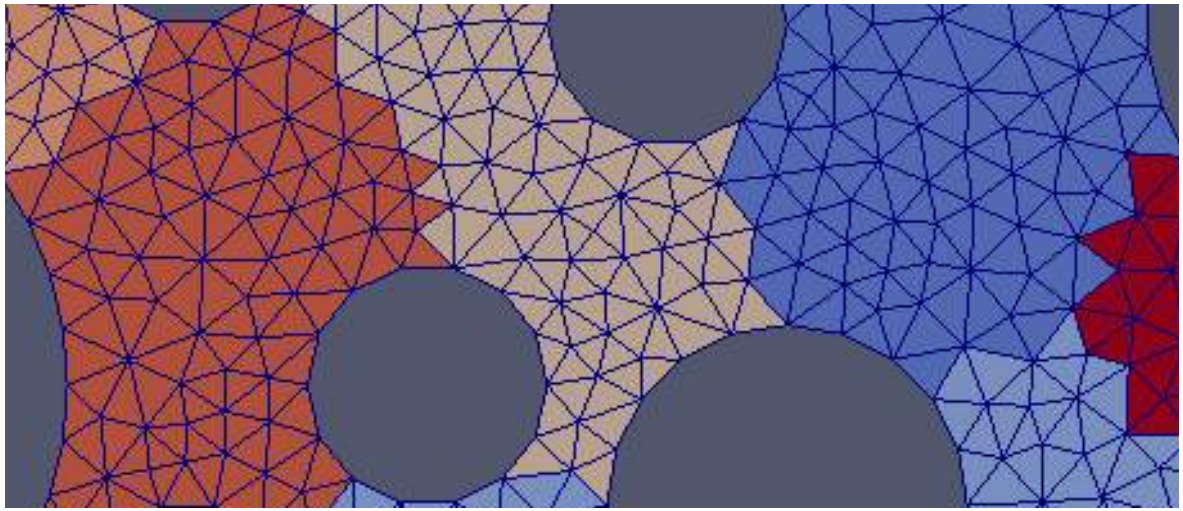


From mesh to primitives (2D)

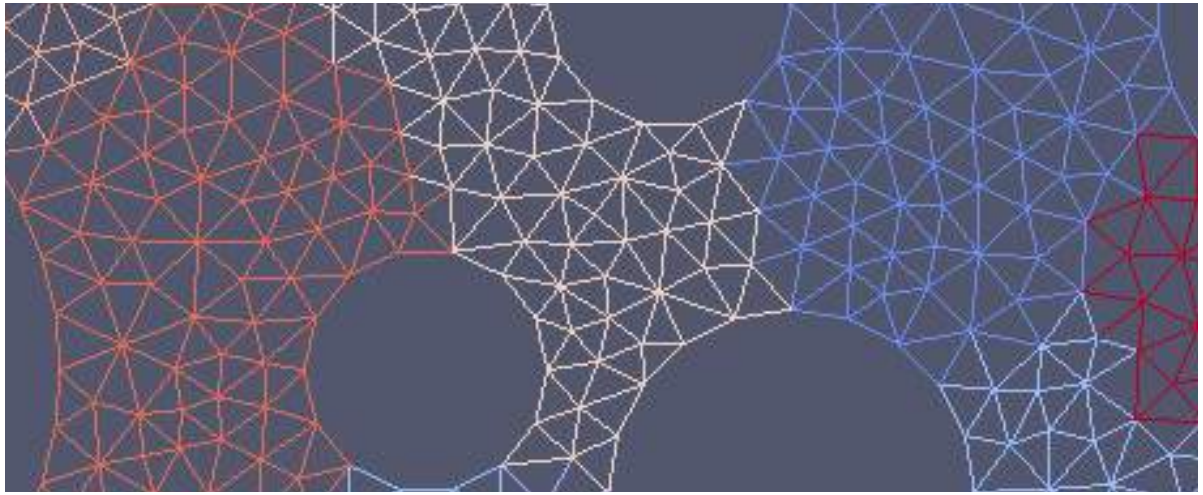


Load Balancing 2D

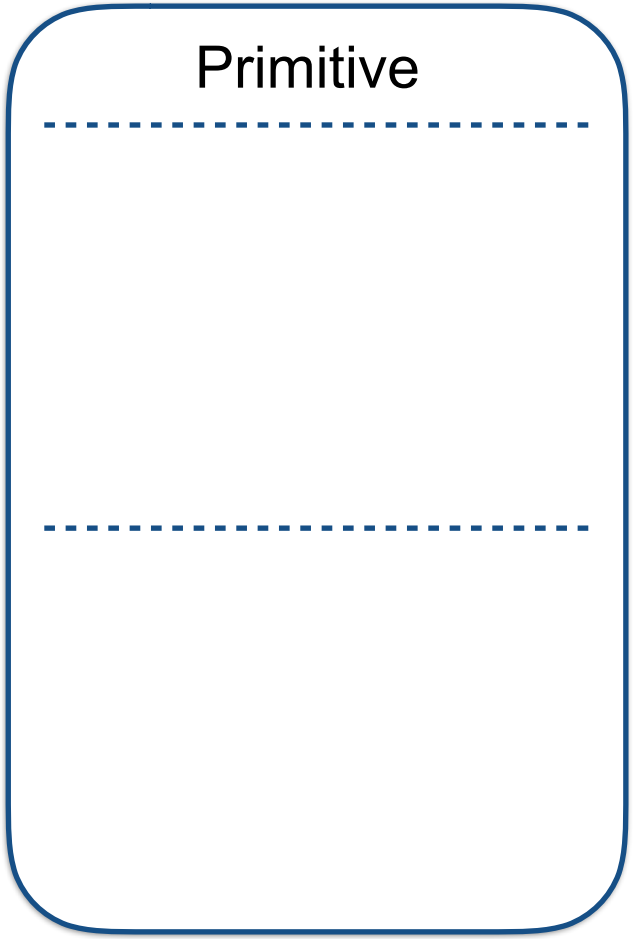
face primitives



edge primitives

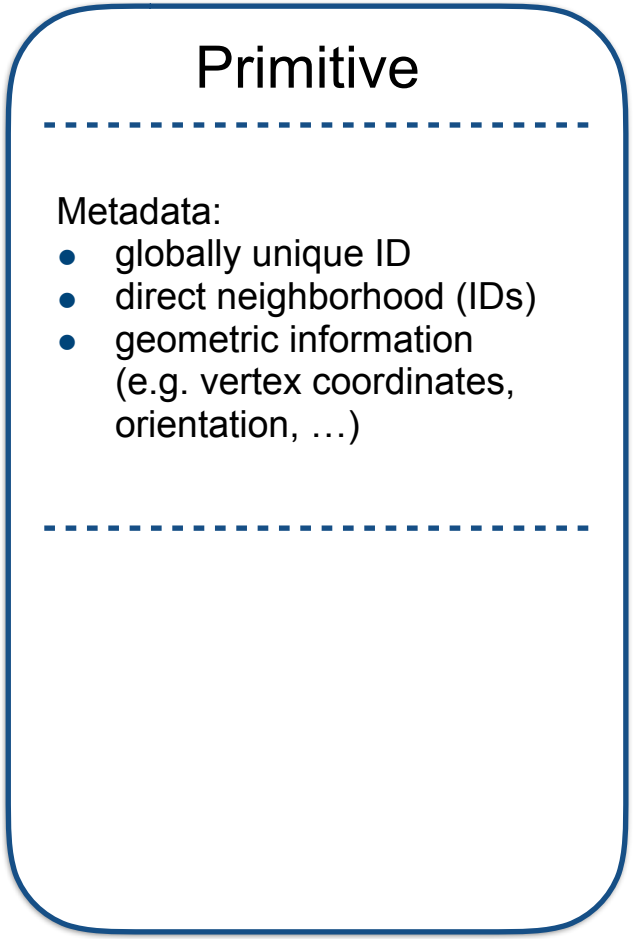


Data Handling



Types: Vertex, Edge, Face, Cell / Tetrahedron

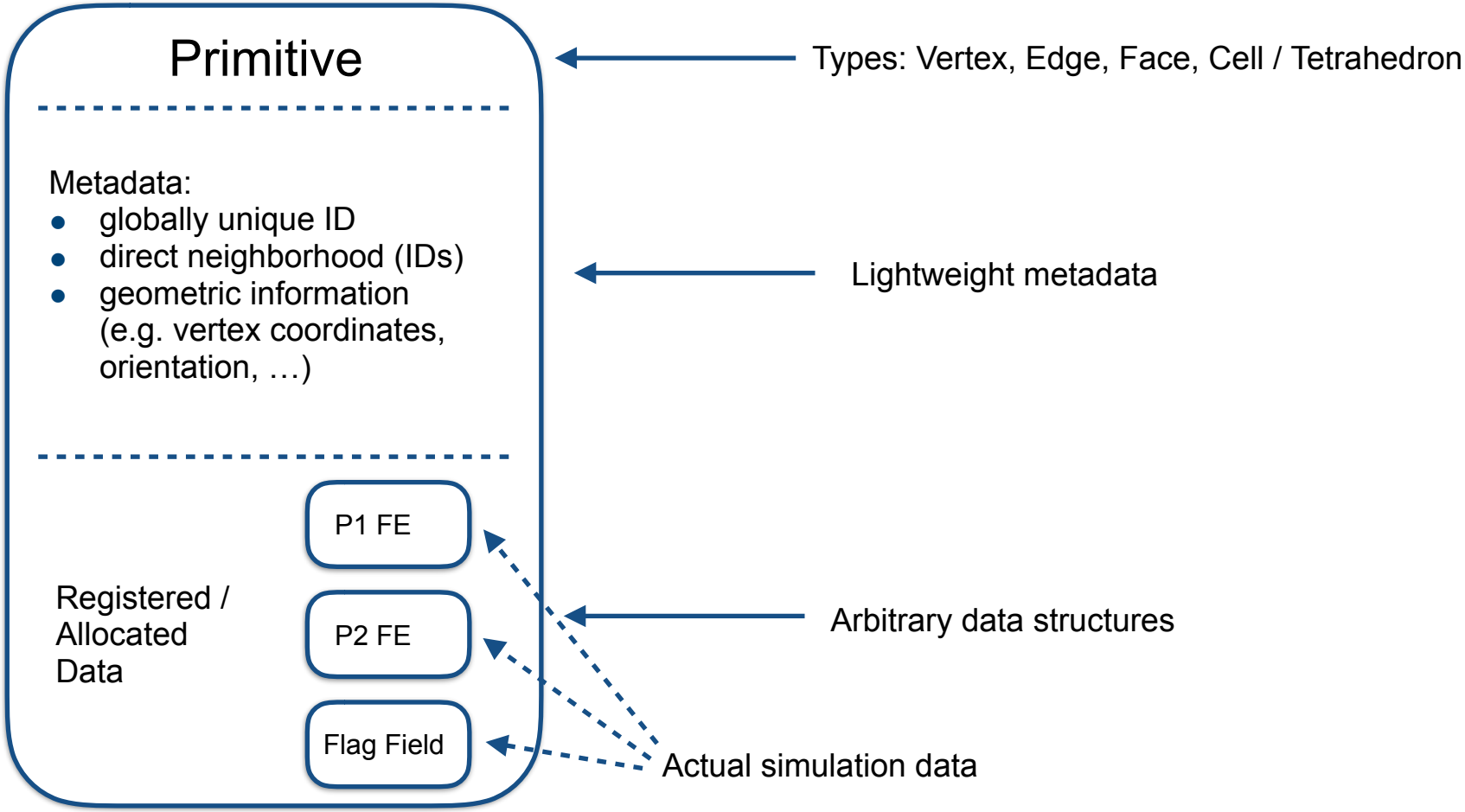
Data Handling



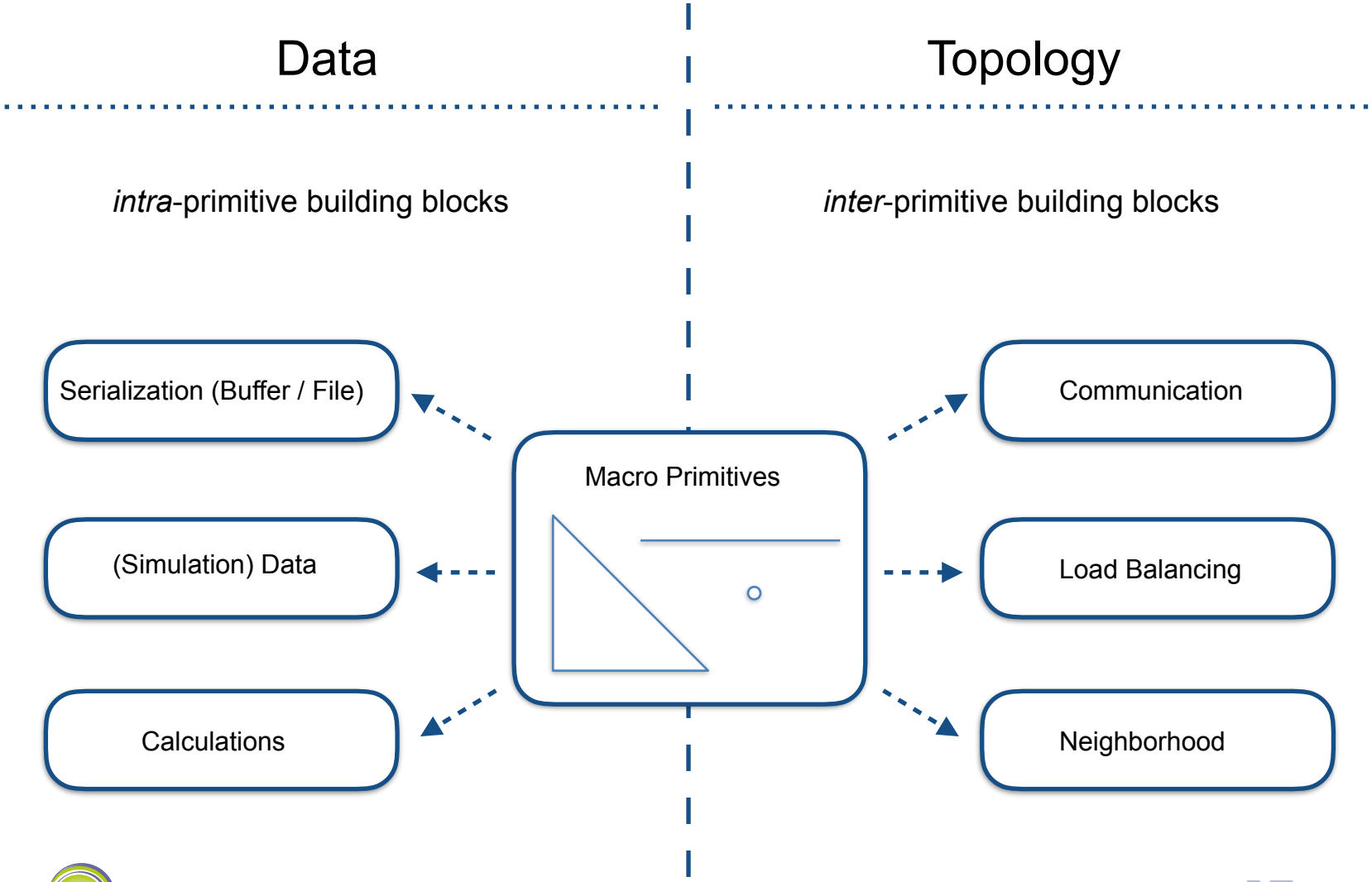
Types: Vertex, Edge, Face, Cell / Tetrahedron

Lightweight metadata

Data Handling



Abstraction Data - Topology



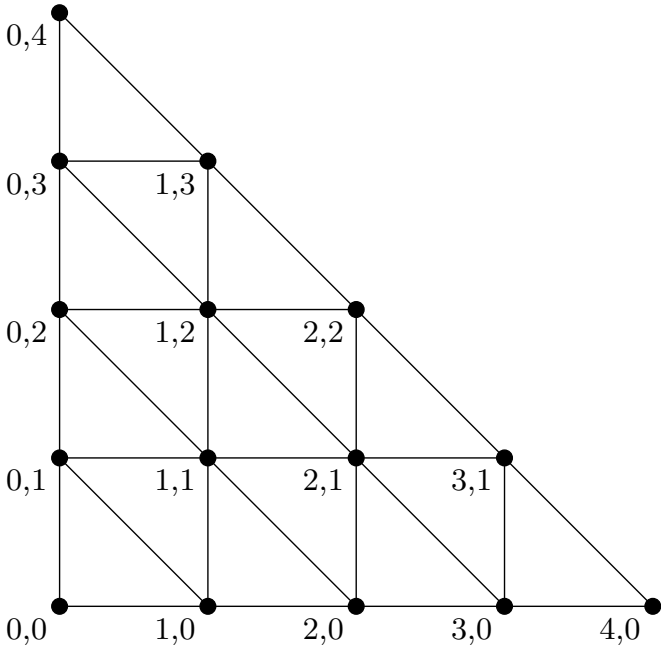
TerraNeo Project

Building Blocks

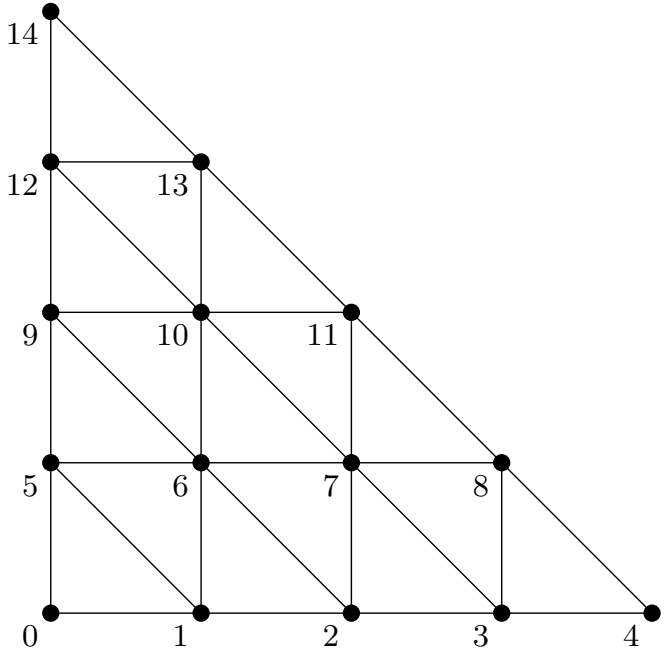
- hierarchical hybrid grids
- fully distributed data structures
- support for different discretizations

Data access abstraction

Abstract index



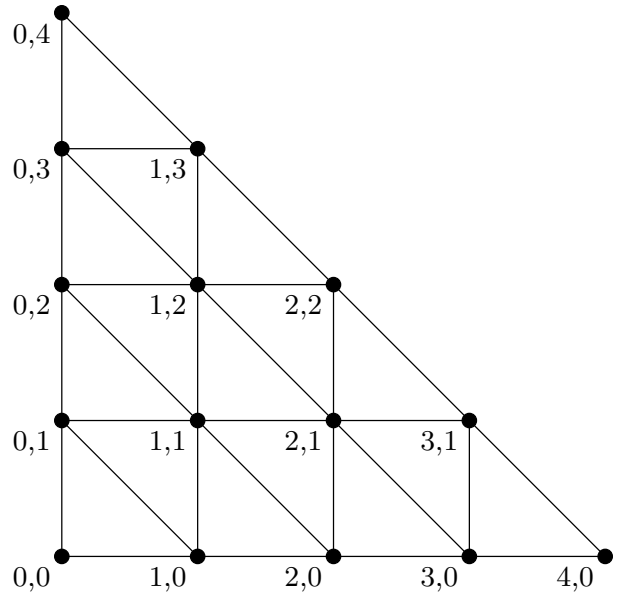
Actual memory index



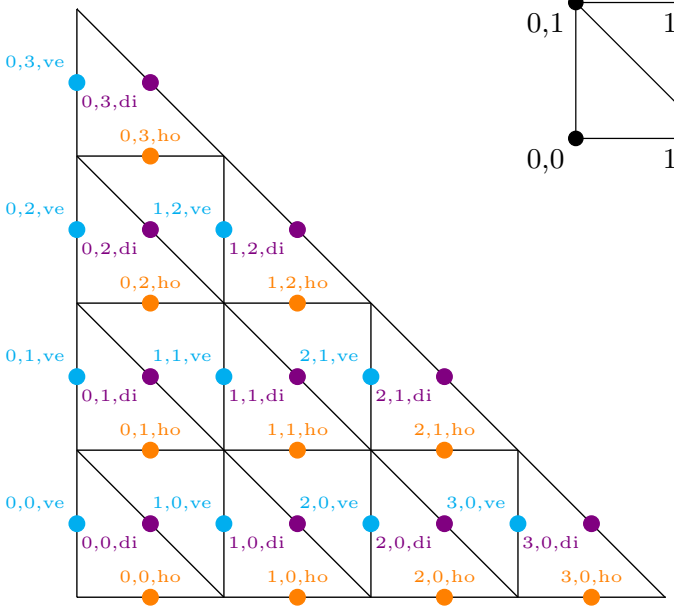

```
face_index(level,x,y) => linearized index

face_index(2    ,3,1) => 8
```

Splitting of unknowns

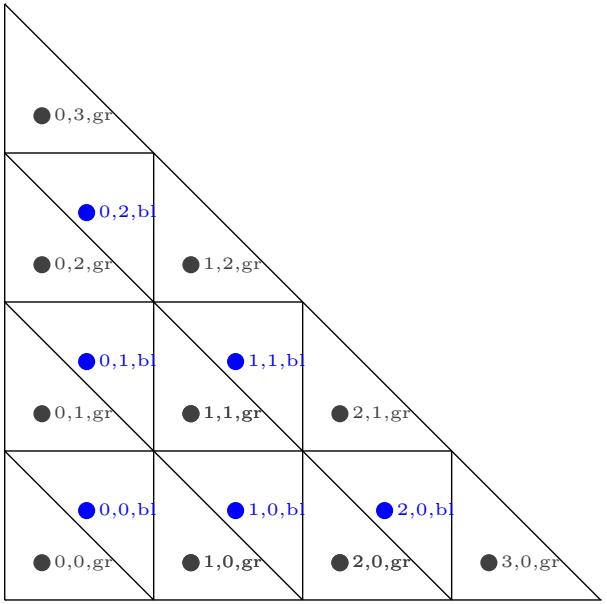
Vertex DoF



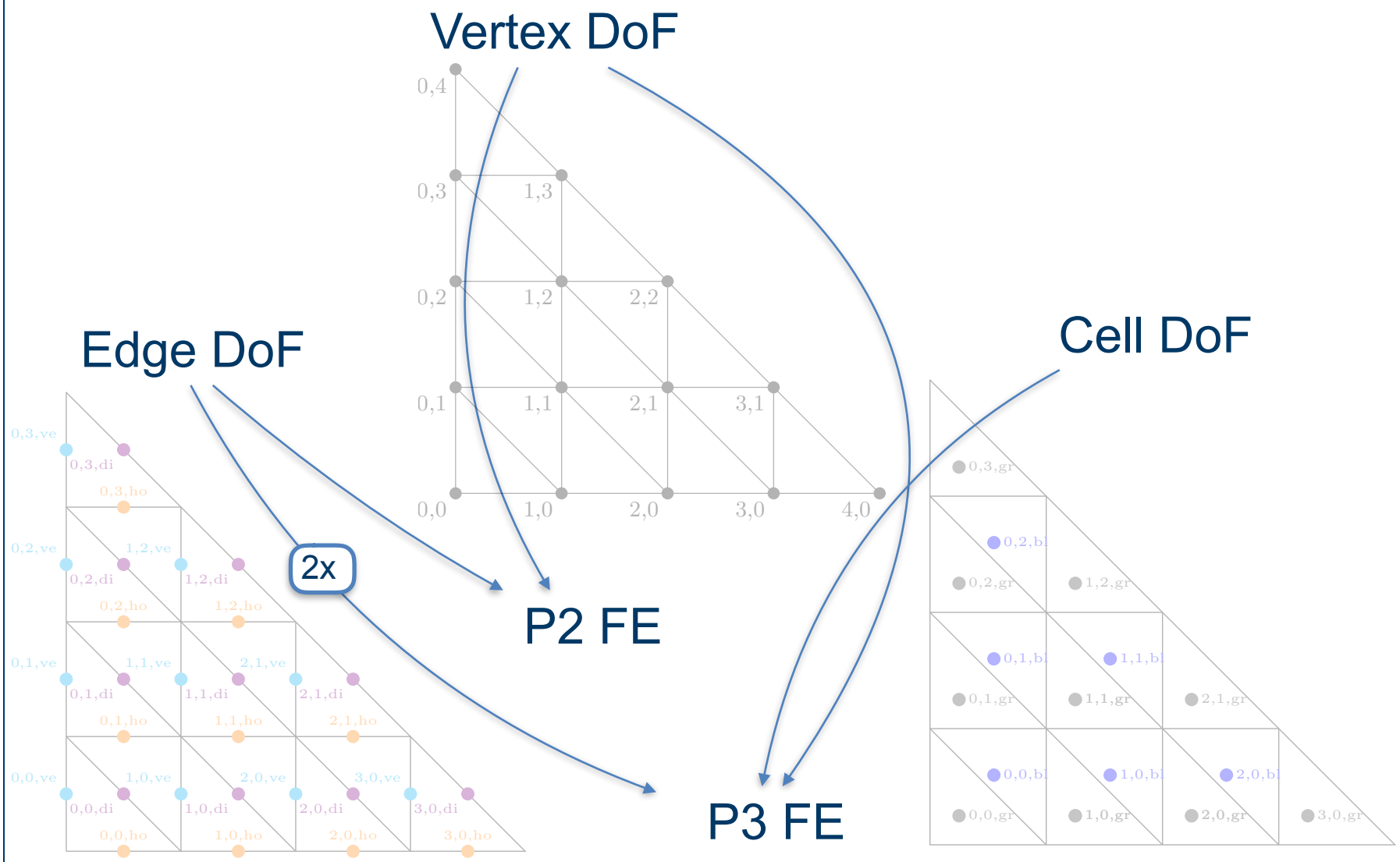
Edge DoF



Cell DoF

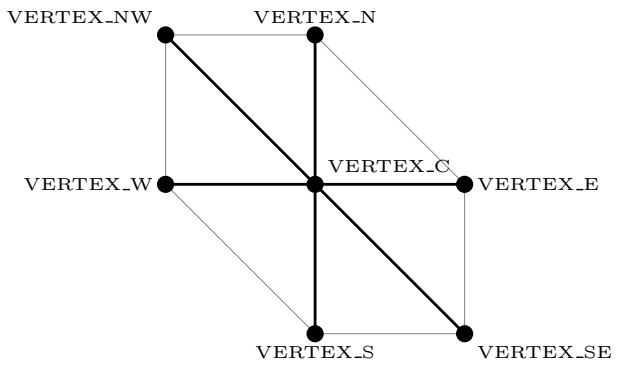


Splitting of unknowns



Outlook: code generation

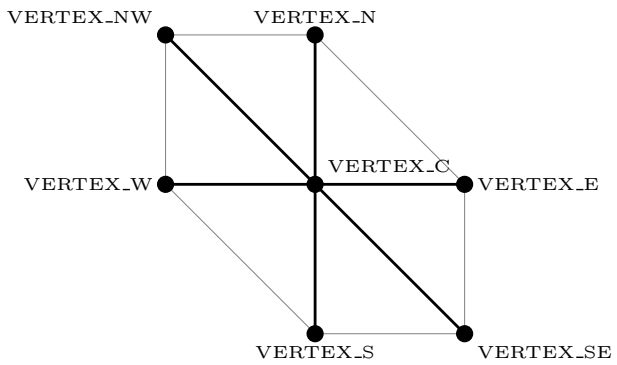
(in communication with ExaStencils)




```
dirs = ['S', 'SE', 'W', 'C', 'E', 'NW', 'N']

src = VertexTriangleField('p1FaceSrc')
dst = VertexTriangleField('p1FaceDst')
stencil = StencilField('p1FaceStencil', dirs)

s = sum([stencil[dir] * src[dir] for dir in dirs])

update = sp.Eq(dst['C'], s)
loop = createTriangleKernel('kernel', [update])
```

Outlook: code generation

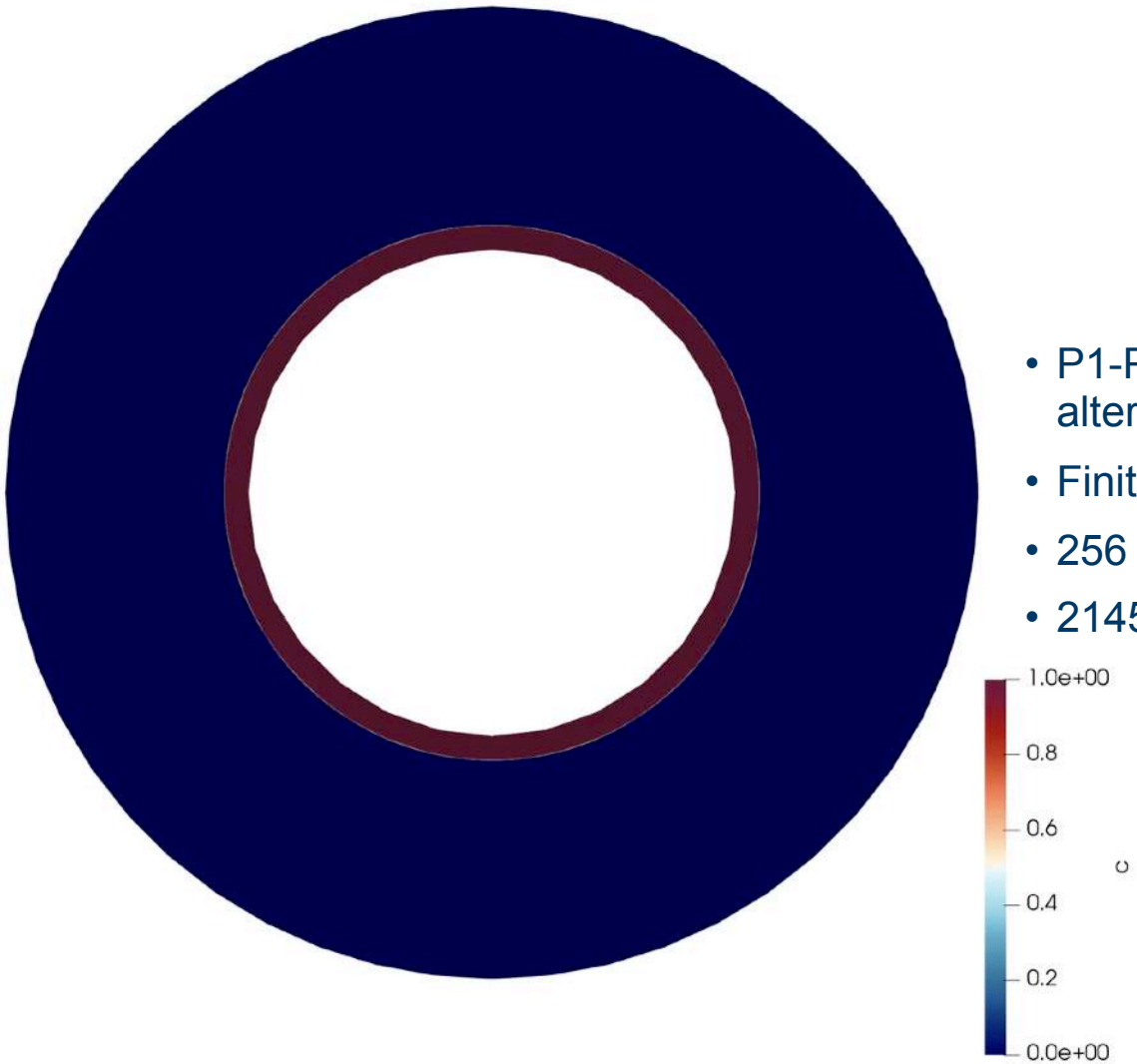
(in communication with ExaStencils)




```
dirs = ['S', 'SE', 'W', 'C', 'E', 'NW', 'N']

src = VertexTriangleField('p1FaceSrc')
dst = VertexTriangleField('p1FaceDst')
stencil = StencilField('p1FaceStencil', dirs)

s = sum([stencil[dir] * src[dir] for dir in dirs])

update = sp.Eq(dst['C'], s)
loop = createTriangleKernel('kernel', [update])
```

```
void kernel(double * fd_p1FaceDst, double * fd_p1FaceSrc, double * fd_p1FaceStencil, int64_t level)
{
   for (int ctr_2 = 1; ctr_2 < (1 << level) - 1; ctr_2 += 1)
   for (int ctr_1 = 1; ctr_1 < -ctr_2 + (1 << level); ctr_1 += 1)
   {
      fd_p1FaceDst[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2)] =
          fd_p1FaceSrc[ctr_1 + (ctr_2 + 1)*((1 << level) + 2) - ((ctr_2 + 1)*(ctr_2 + 2) / 2) - 1]*fd_p1FaceStencil[5]
        + fd_p1FaceSrc[ctr_1 + (ctr_2 + 1)*((1 << level) + 2) - ((ctr_2 + 1)*(ctr_2 + 2) / 2)]*fd_p1FaceStencil[6]
        + fd_p1FaceSrc[ctr_1 + (ctr_2 - 1)*((1 << level) + 2) - (ctr_2*(ctr_2 - 1) / 2) + 1]*fd_p1FaceStencil[1]
        + fd_p1FaceSrc[ctr_1 + (ctr_2 - 1)*((1 << level) + 2) - (ctr_2*(ctr_2 - 1) / 2)]*fd_p1FaceStencil[0]
        + fd_p1FaceSrc[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2) + 1]*fd_p1FaceStencil[4]
        + fd_p1FaceSrc[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2) - 1]*fd_p1FaceStencil[2]
        + fd_p1FaceSrc[ctr_1 + ctr_2*((1 << level) + 2) - (ctr_2*(ctr_2 + 1) / 2)]*fd_p1FaceStencil[3];
   }
}
```

Annulus convection



- P1-P1 PSPG Elements for Stokes alternatively Taylor-Hood
- Finite volumes for heat transport
- 256 face primitives
- 2145 DoFs on each face p. (level 6)