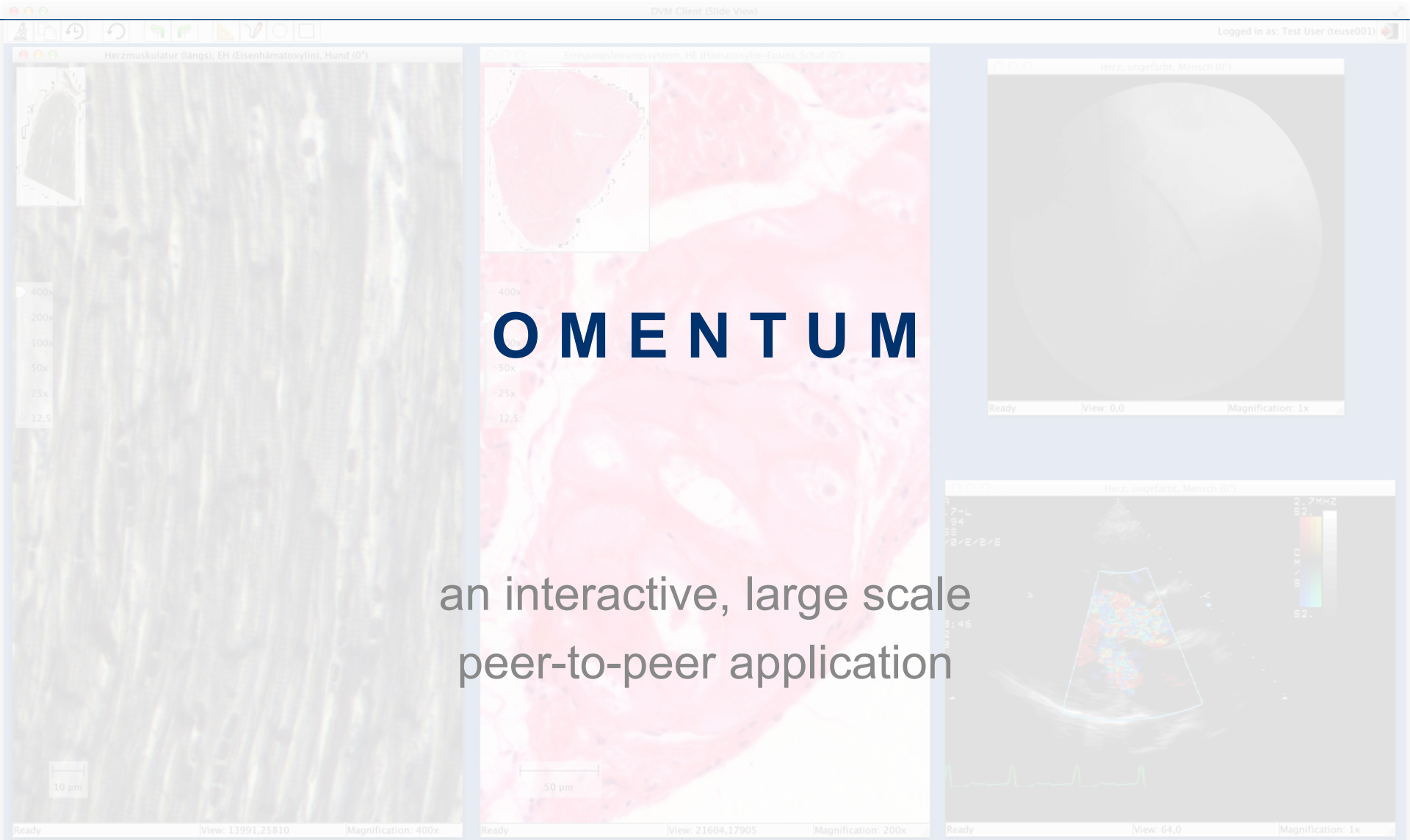
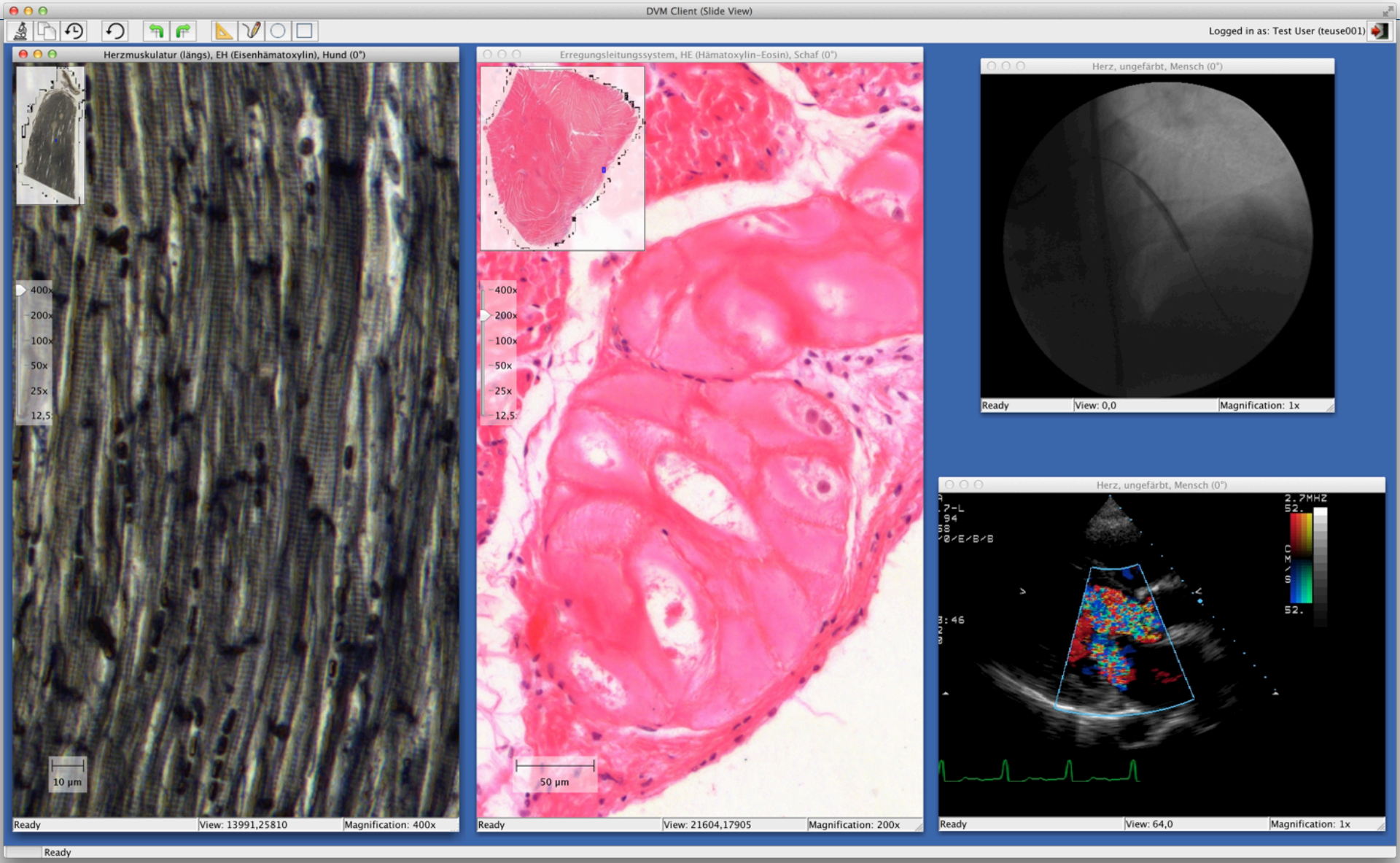


OMENTUM

an interactive, large scale
peer-to-peer application

Andreas Barbian
Departments of Anatomy and Computer Science


DVM Client (Slide View)
Logged in as: Test User (teuse001)
Herzmuskulatur (längs), EH (Eisenhämatoxylin), Hund (0°)
400x
200x
100x
50x
25x
12,5
10 µm
Ready
View: 13991,25810
Magnification: 400x
Erregungsleitungssystem, HE (Hämatoxylin-Eosin), Schaf (0°)
400x
200x
100x
50x
25x
12,5
50 µm
Ready
View: 21604,17905
Magnification: 200x
Herz, ungefärbt, Mensch (0°)
Ready
View: 0,0
Magnification: 1x
Herz, ungefärbt, Mensch (0°)
2.7MHZ
Ready
View: 64,0
Magnification: 1x
Ready

Virtual Microscopy

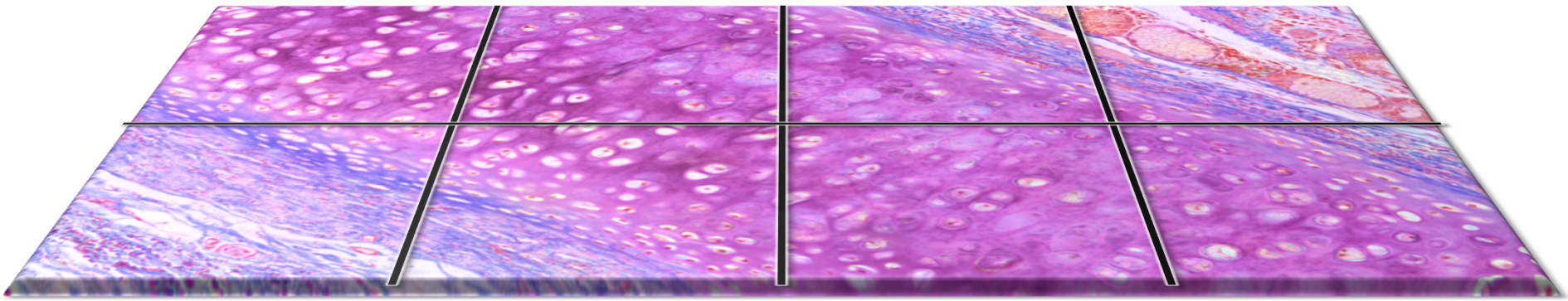
HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

- virtual slides
  - large and mostly proprietary files (~ 50 GB / slide)
  - 150.000 x 260.000 pixel
- textual (or audio/video) annotations in the slide
  - creation and maintenance (e.g. collections > 5000 slides)
- location-independent learning
  - more than laboratory sized access groups
  - infrastructure
  - application
    - license and library restrictions

Tile Computing

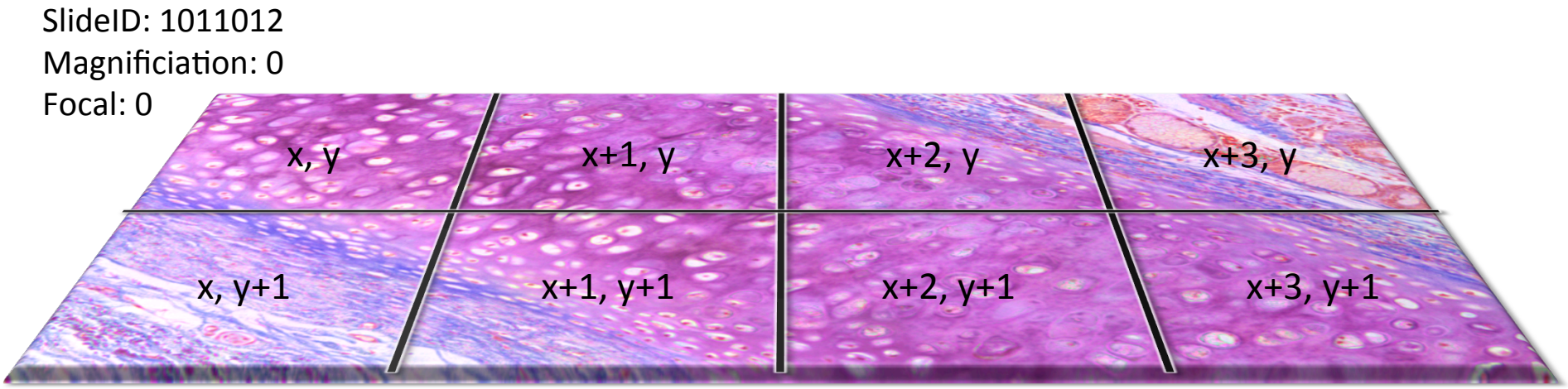
- pre-computing
  - no need for realtime extraction
  - keeping files small
  - retrieving files from multiple sources at once
  - easily more than 10^5 files per layer

```
computeTile(x, y, magLevel, fLayer, width, height)
```



Tile Computing

- tiles are unique and identified by
  - SlideID
  - Magnification Level
  - Focal Plane
  - Coordinates



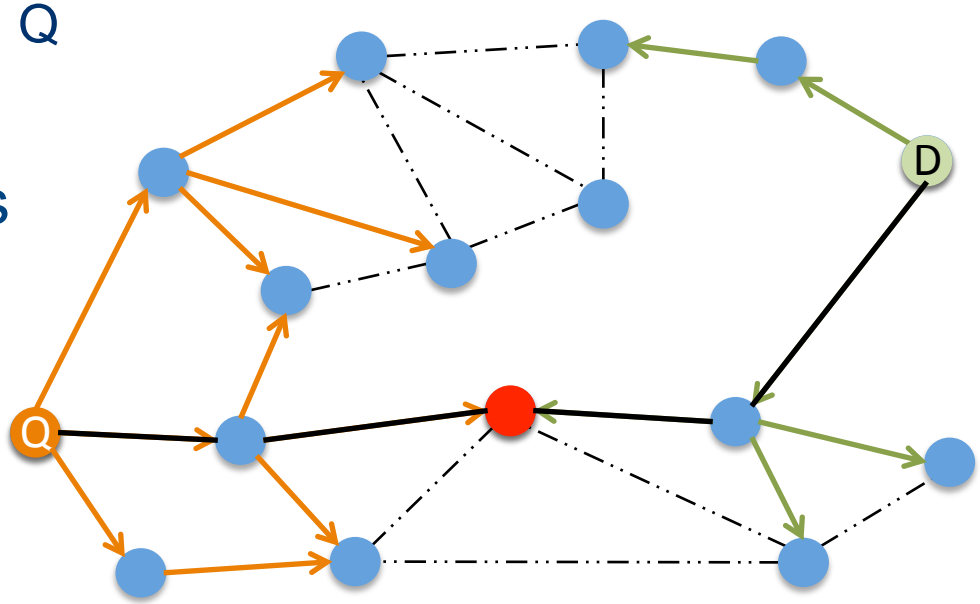
Strategy for Tile Location

- slide information is distributed (e.g. width & height)
- tiles are grouped in partitions
  - number of partitions can be calculated from dimension
- each partition represents a virtual node
  - aggregate partition number over all slide with lower ID => vNodeID

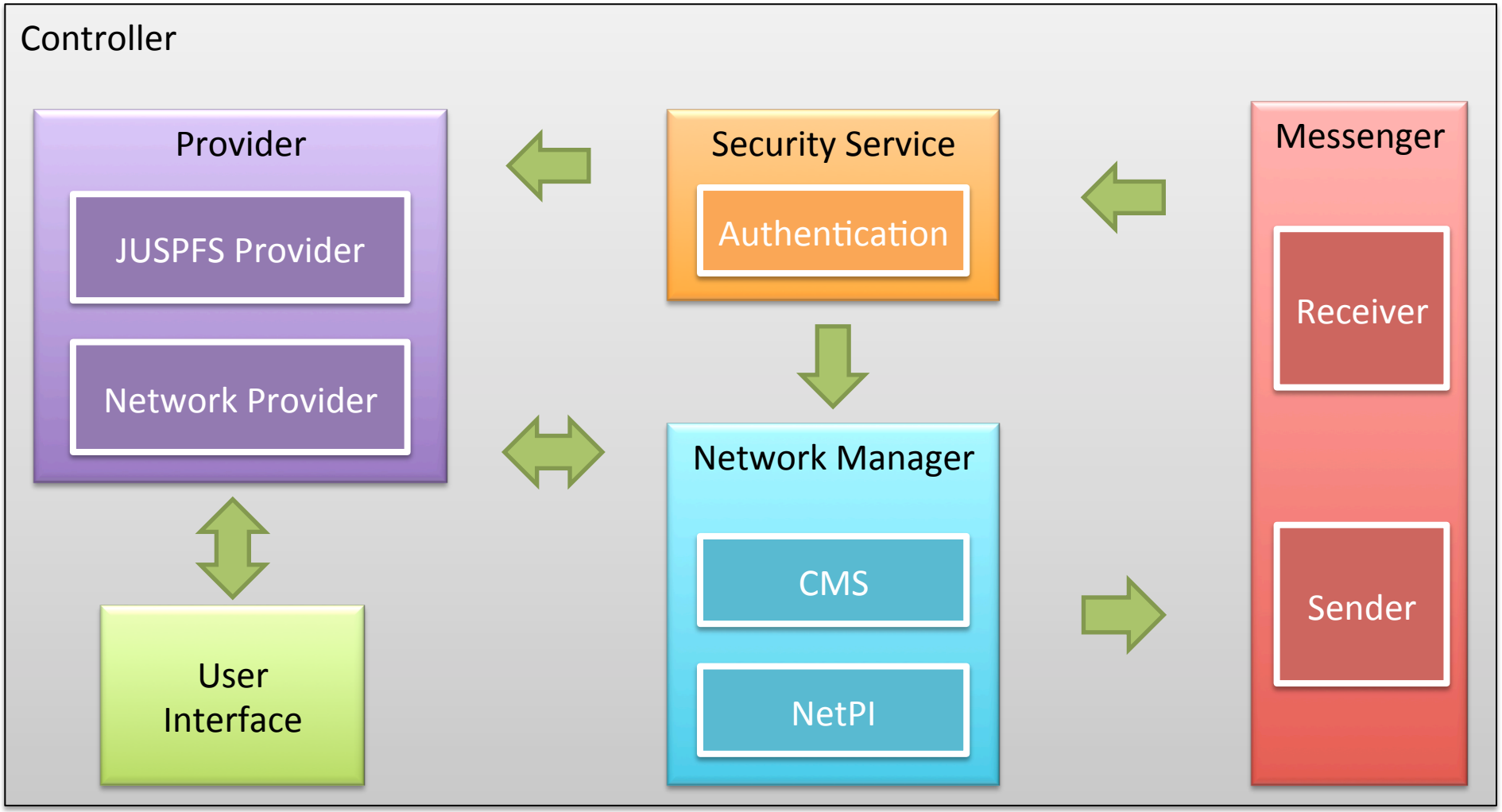
- locating tile
  - get ID of requested slide
  - calculate partition the tile belongs to
  - aggregate partition numbers for “earlier” slides to get vNodeID
  - locally calculate routing path to virtual node holding partition
  - route to vNode and request tile

Object Retrieval

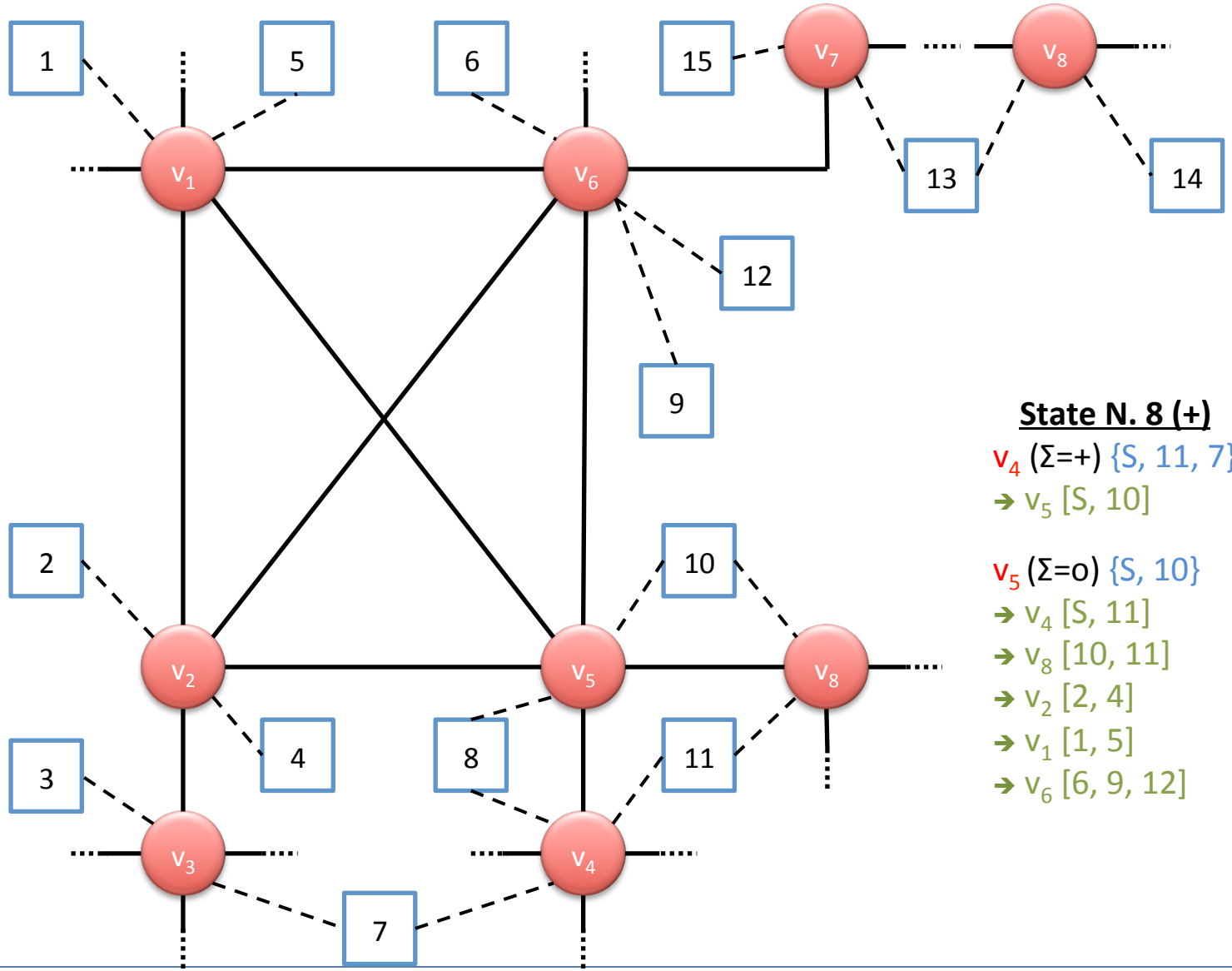
- 2 PRNGs used to create random graph structure
  - determine number of neighbors
  - determine IDs of neighbors
- path from Q to D without network traffic
  - compute partition
  - determine neighbors of D and Q
- routing via calculated nodes
- based on Pathfinder



Omentum: Architecture



Load Balancing & Performance Index



State N. 8 (+)

v_4 (Σ=+) {S, 11, 7}
➔ v_5 [S, 10]

v_5 (Σ=o) {S, 10}
➔ v_4 [S, 11]
➔ v_8 [10, 11]
➔ v_2 [2, 4]
➔ v_1 [1, 5]
➔ v_6 [6, 9, 12]

Node State	
11	+
10	-
2	o
4	-
7	+
1	+
5	o
6	-
9	+
12	o