ROI Model Draft Overview

Roger Leigh

6th November 2012
University of Dundee

Overview

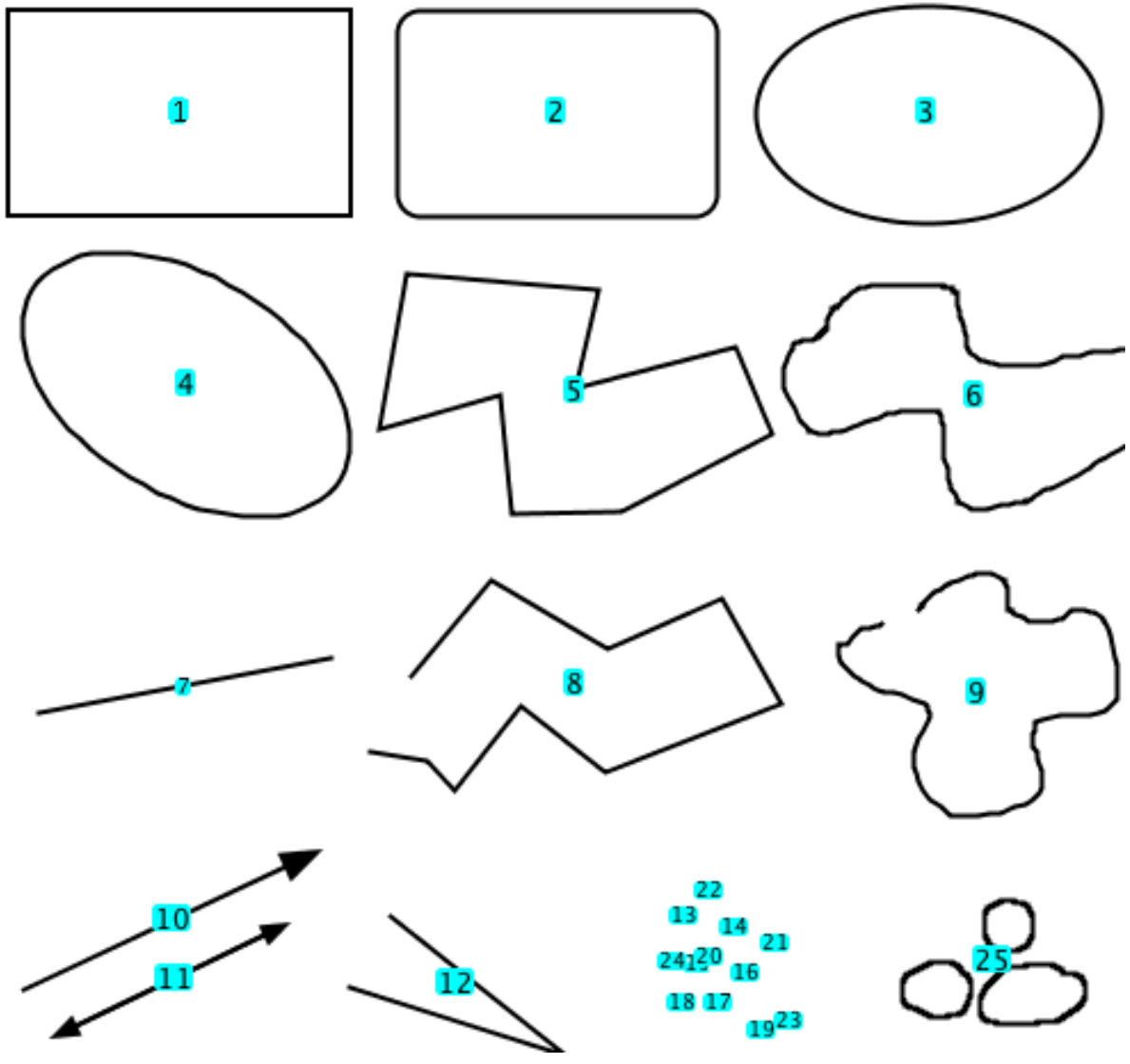
- Existing models
- Model draft
- Shapes
  - Example: A sphere
  - Shapes and representations
  - Shape serialisation
  - Shape relationships
  - Nested/stacked shapes and transformations
- Masks
- Drawing
- Editing
- Further discussion

What is a ROI?

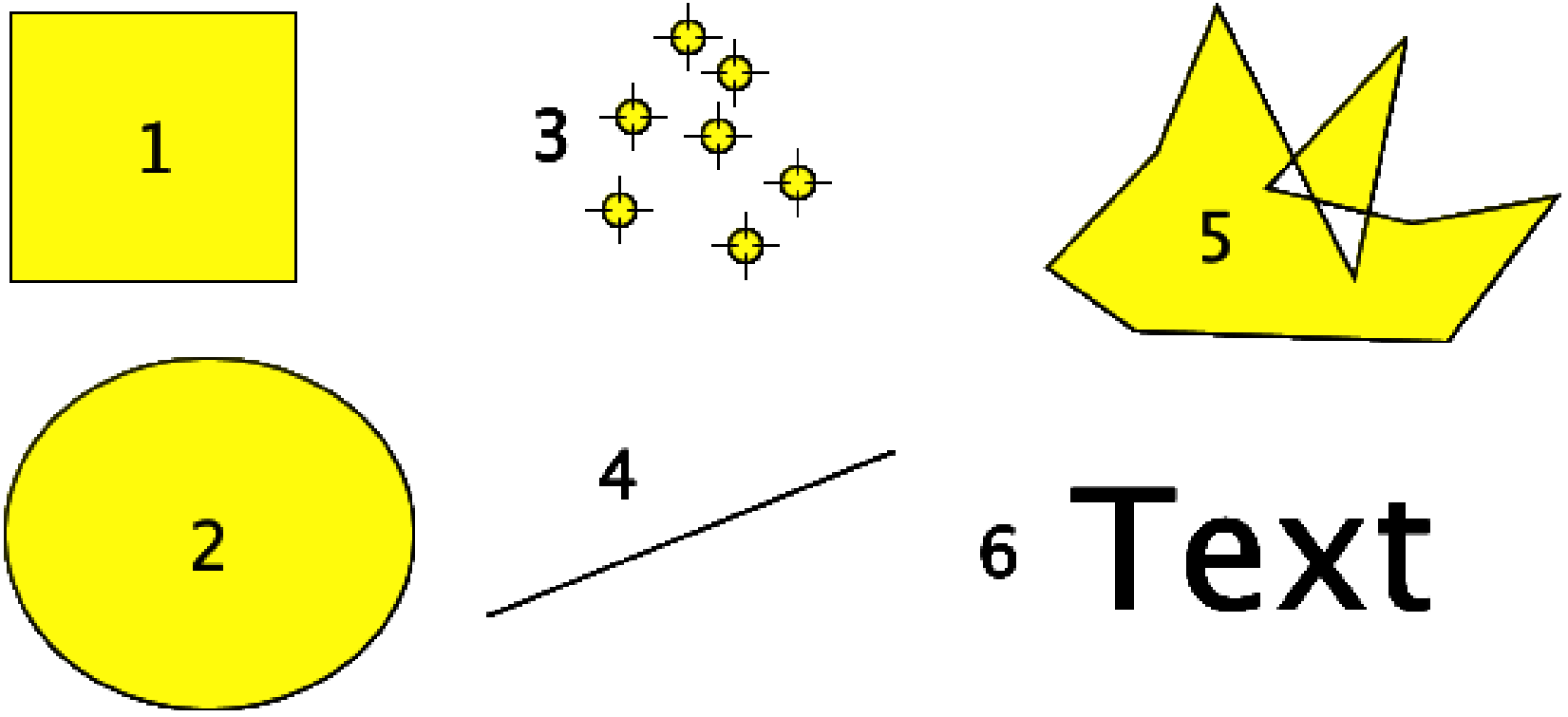
ROI Region of interest. A subset of samples within an image. This is specified by the boundary or surface of the object.

Shape Geometric shape or mask. A shape is a geometric primitive or bitmask. A ROI is composed of one or more shapes.

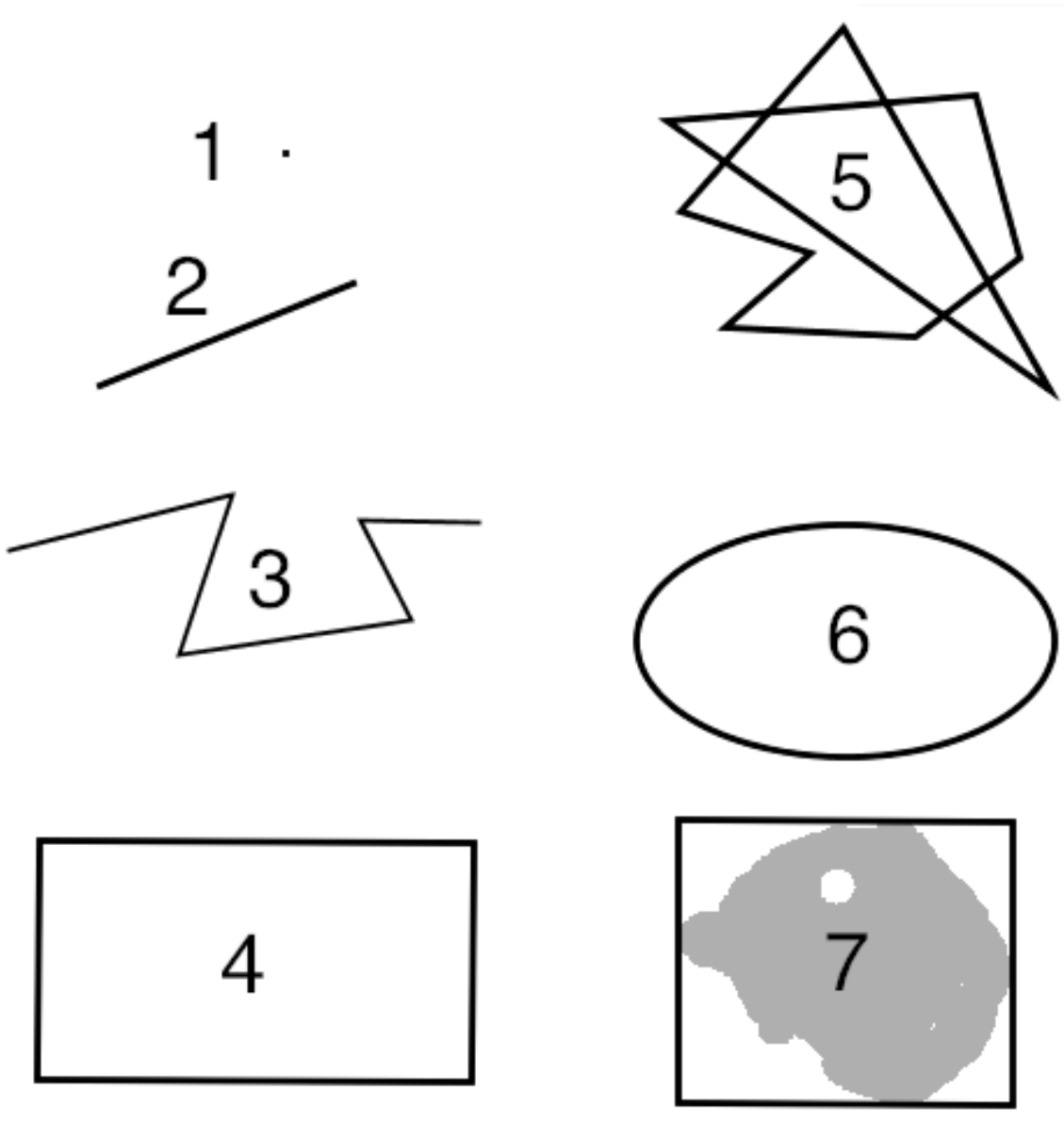
Existing models: ImageJ



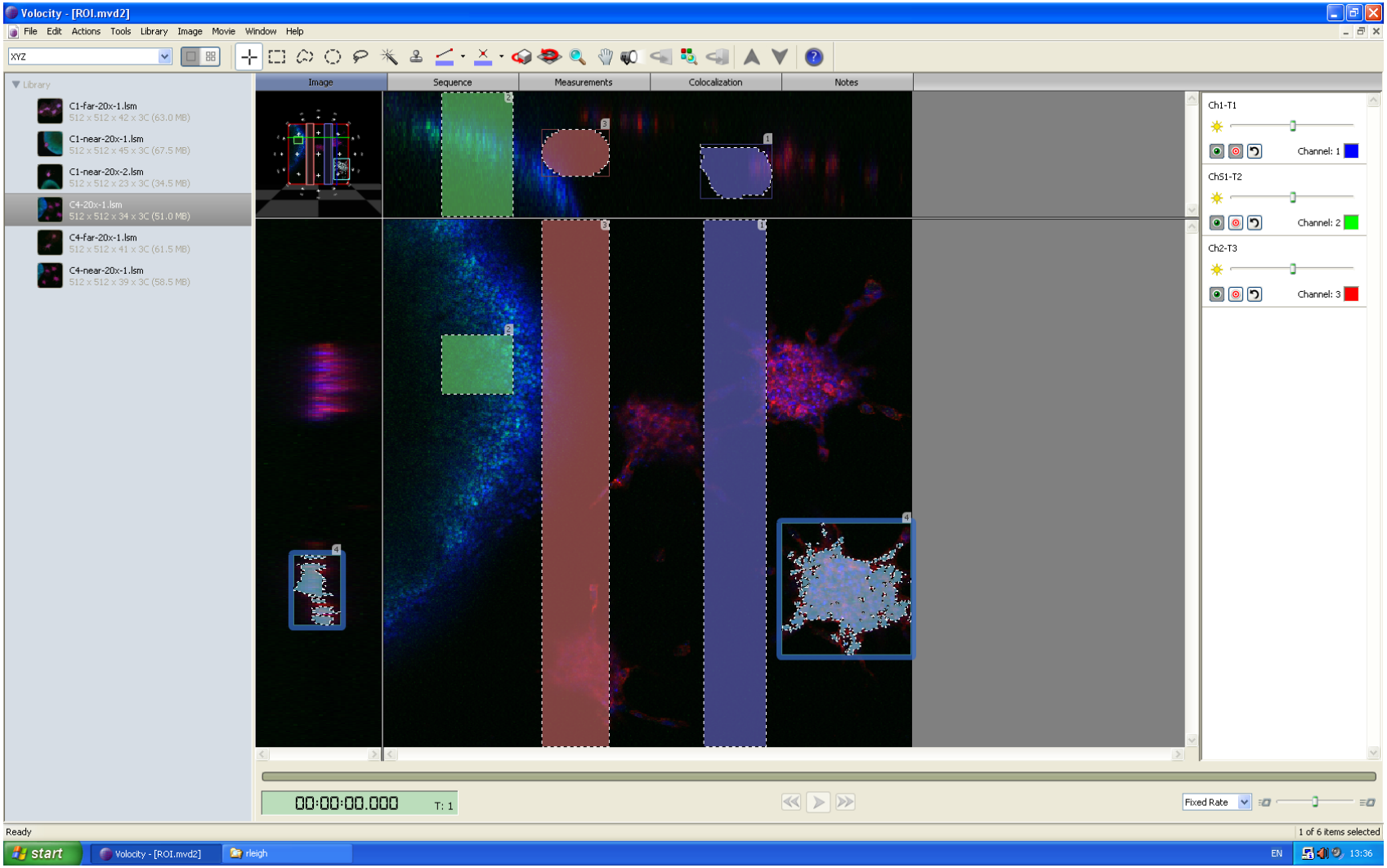
Existing models: Insight



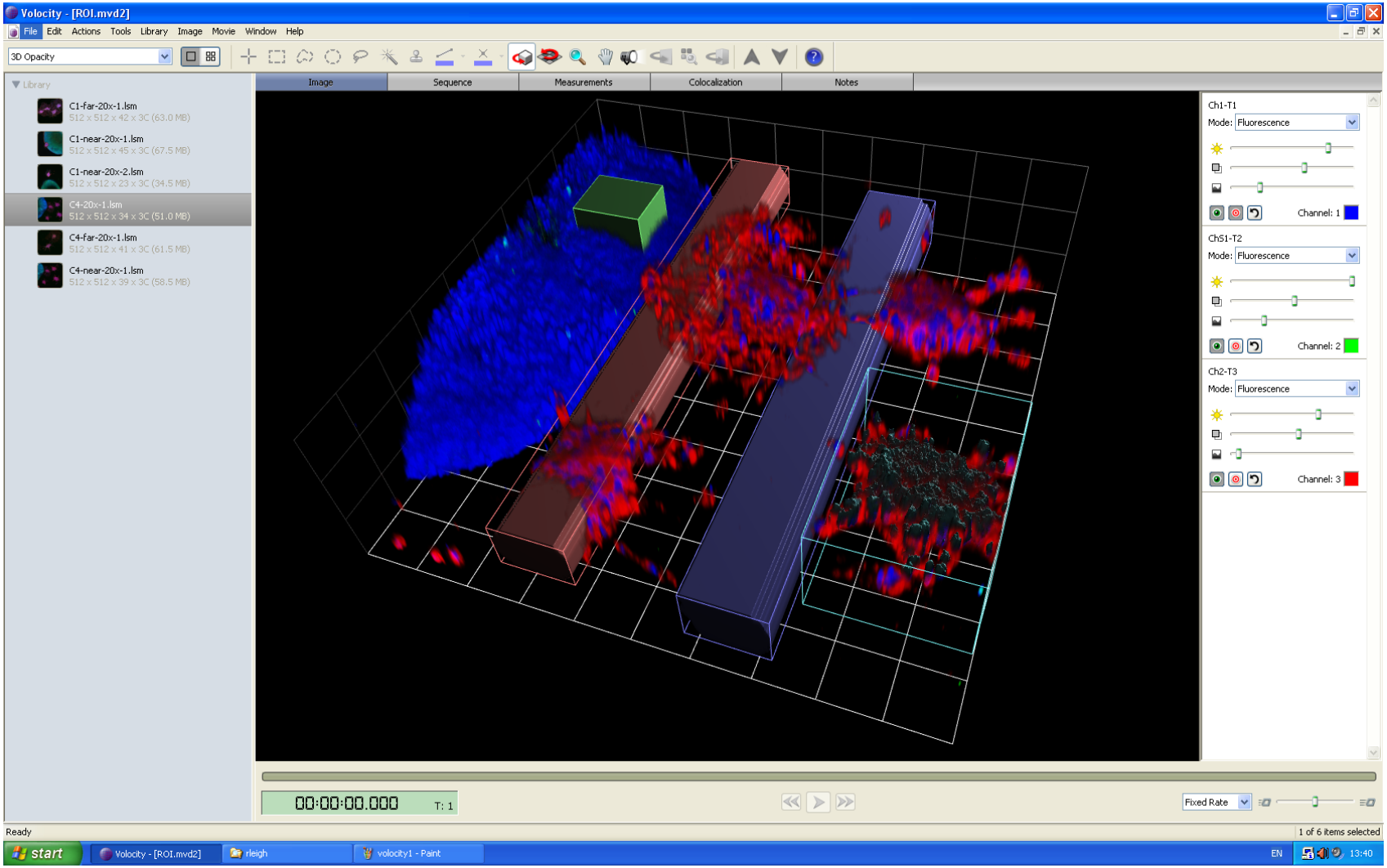
Existing models: Icy



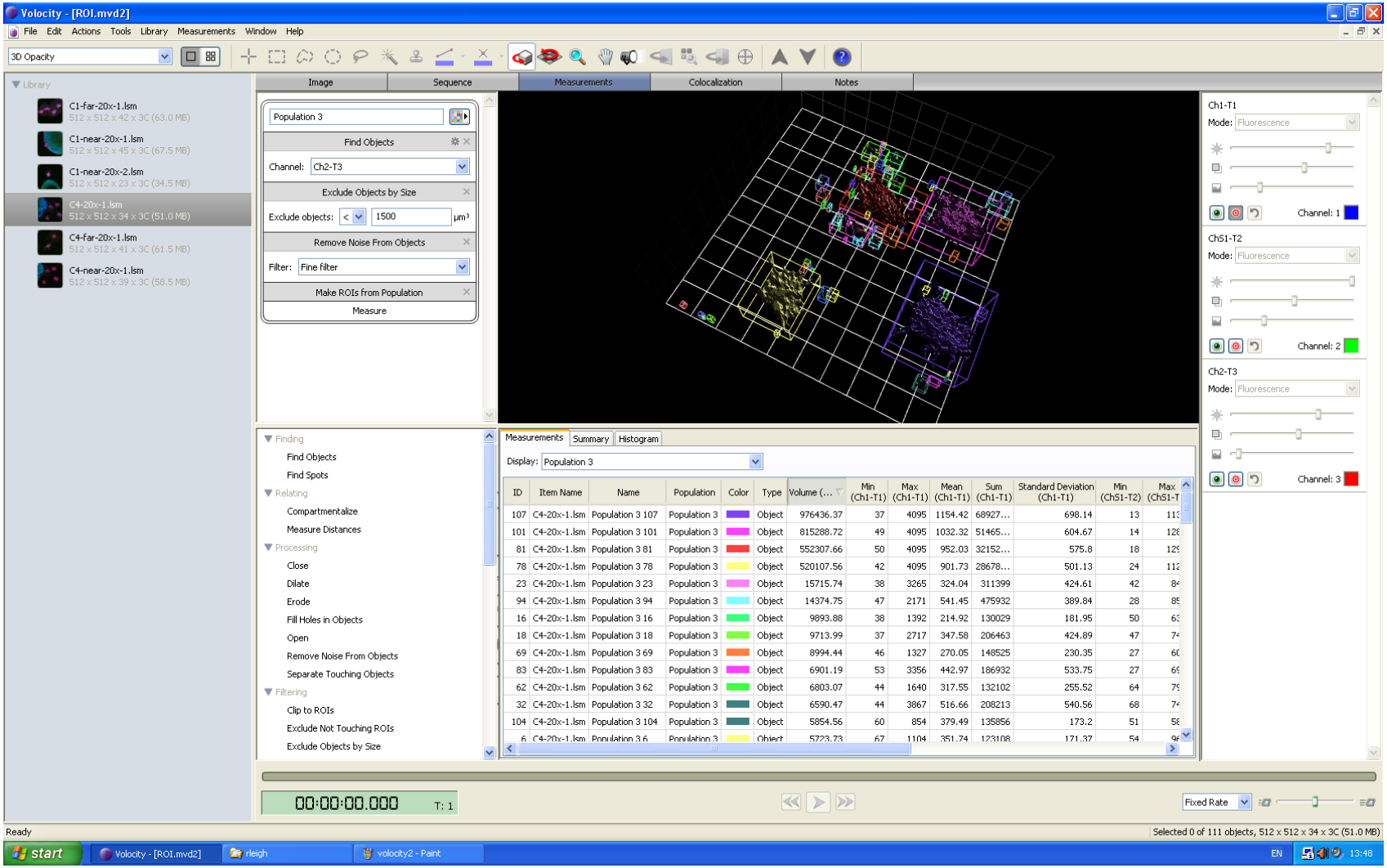
Existing models: Volocity



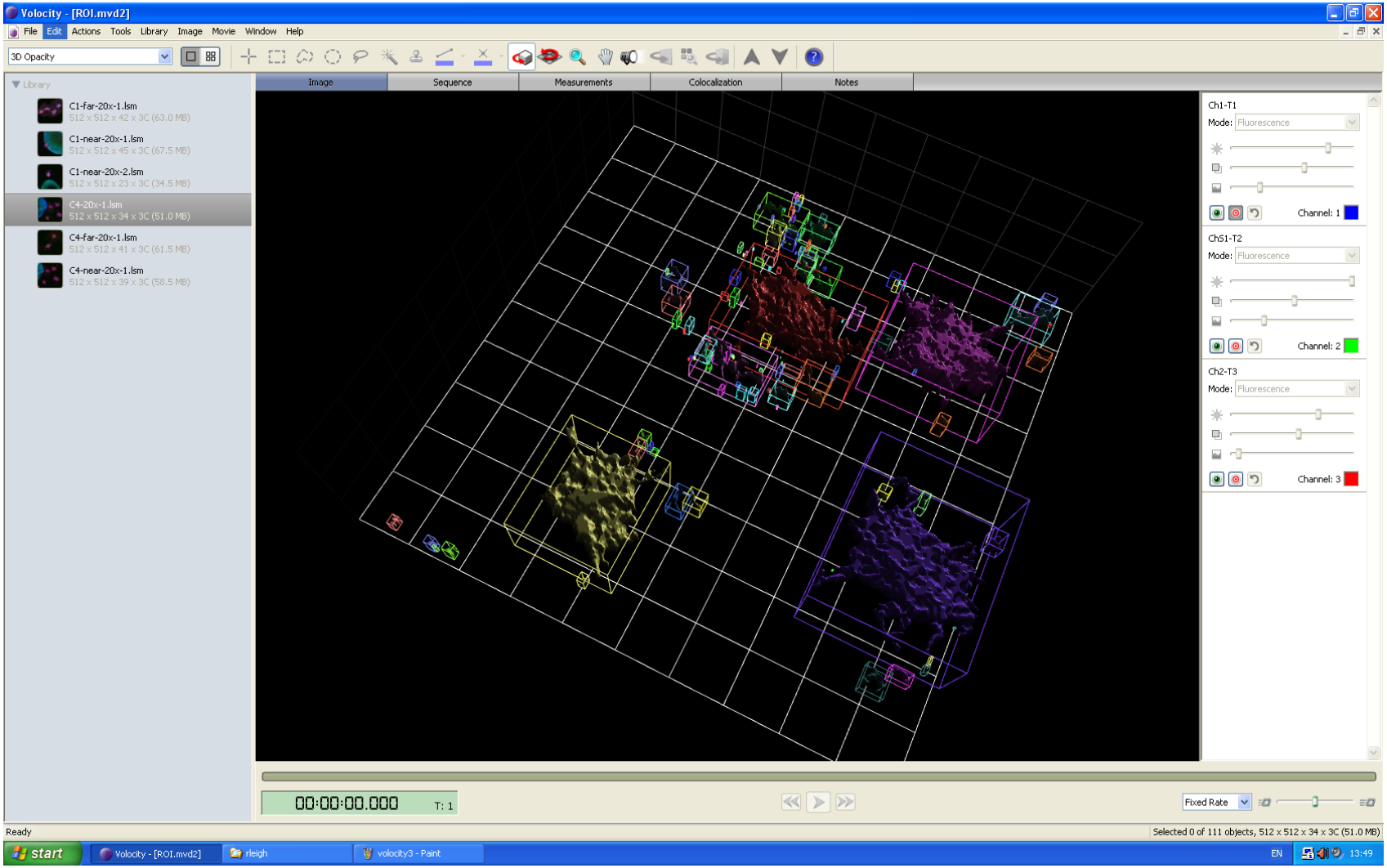
Existing models: Volocity



Existing models: Volocity



Existing models: Volocity



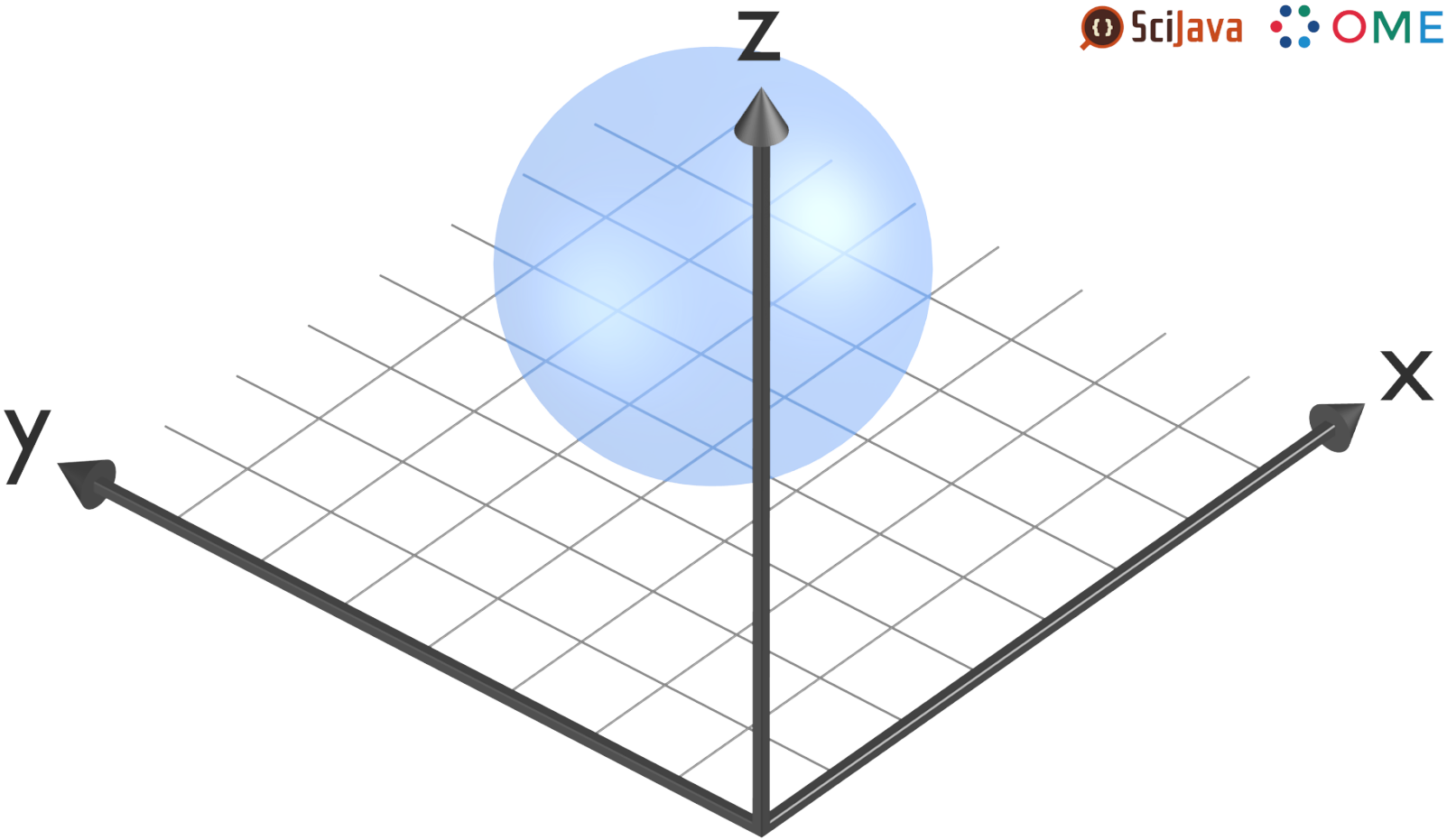
The draft specification

- This is a work in progress
- Everything is changeable, nothing is fixed
- In git
  - `git://github.com/scijava/roi-model.git`
- Specification text
  - Sphinx markup
  - `*.rst`
- Storage/interface definitions
  - Tab-separated tabular data
  - `spec/*.txt`
- Code/specification generator
  - `genspec`, `python/*.py`
- Java/C++/other reference implementations (TBD)

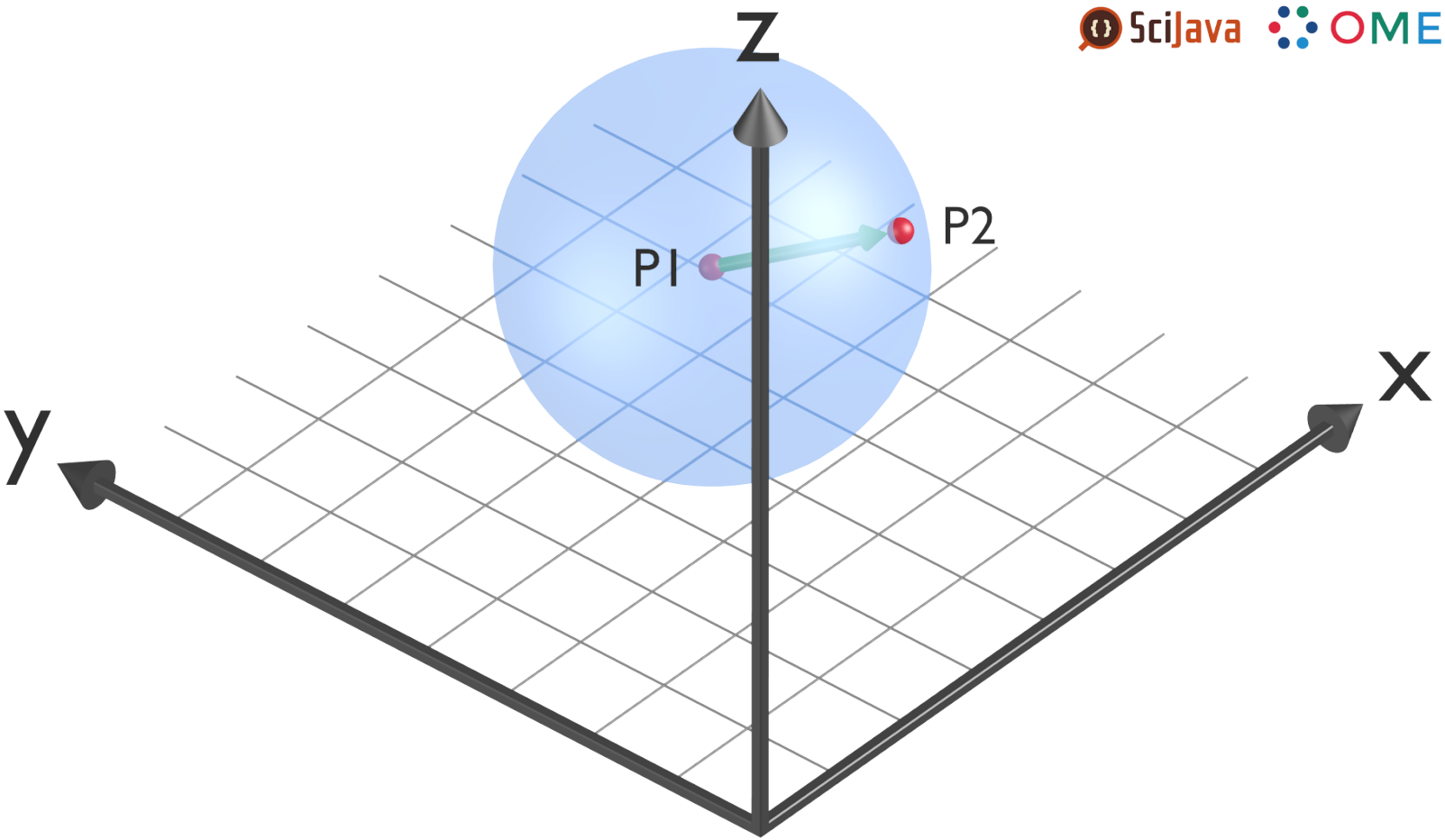
The draft specification

- This specification addresses:
  - Describing ROIs
  - Serialising ROIs for storage and exchange
  - Converting ROIs to iterable entities
  - Drawing ROIs
  - Editing ROIs
- This specification *does not* address:
  - ROI-ROI links for tracking, and other high-level ROI inter-relationships.
  - A directed graph of ROI-ROI links would be a potential solution.

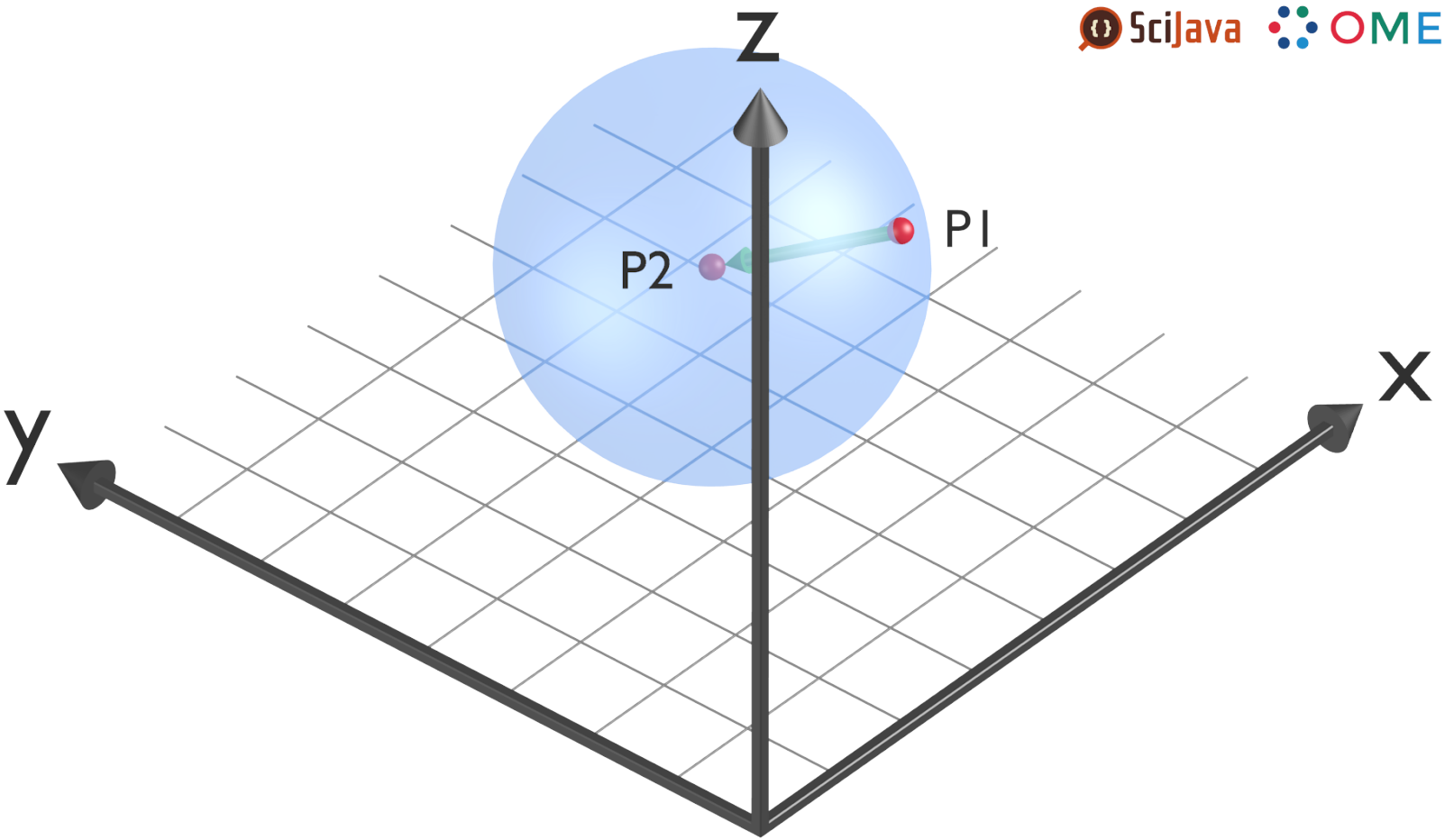
Describing a sphere



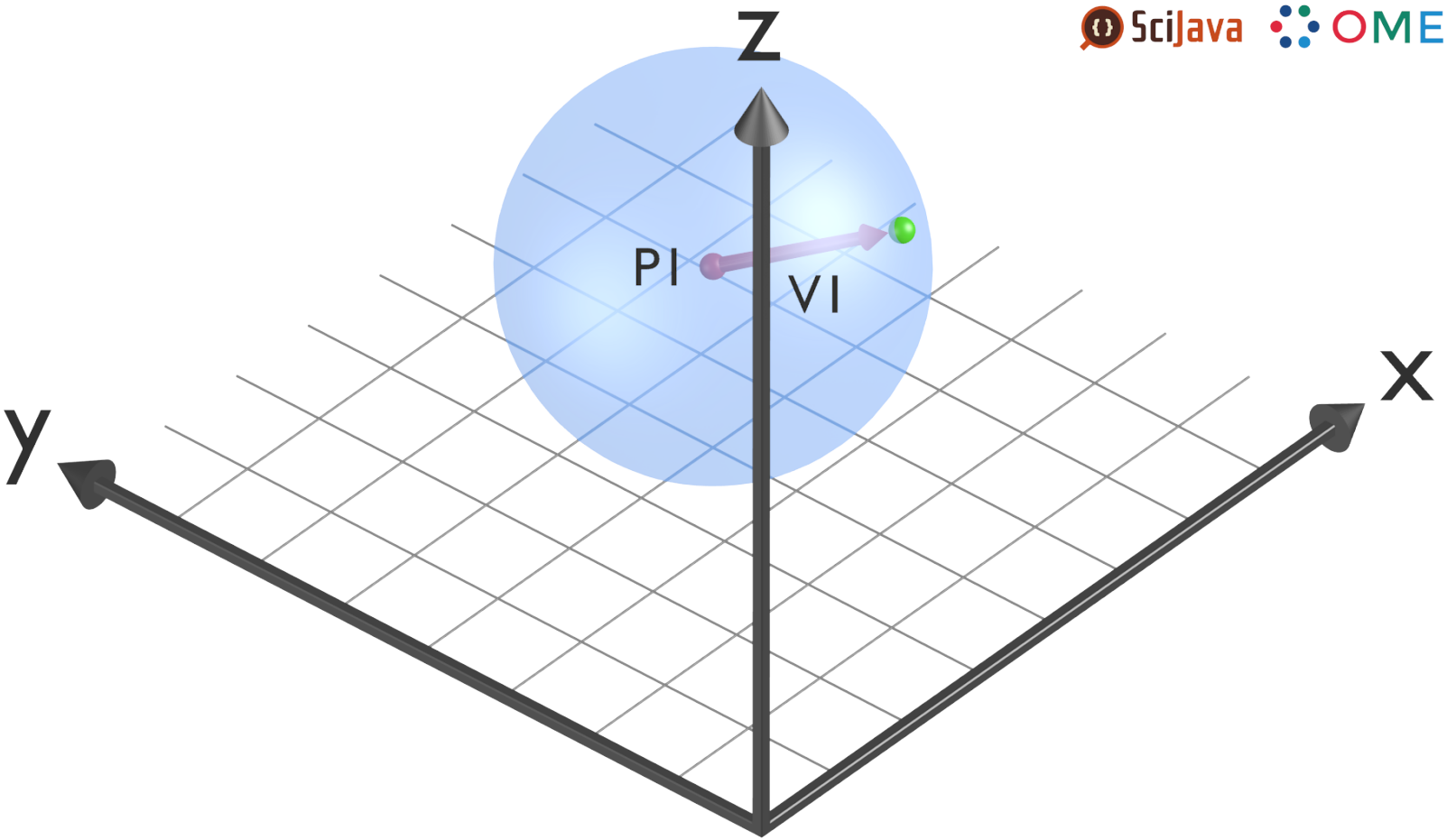
Describing a sphere: centre and radius



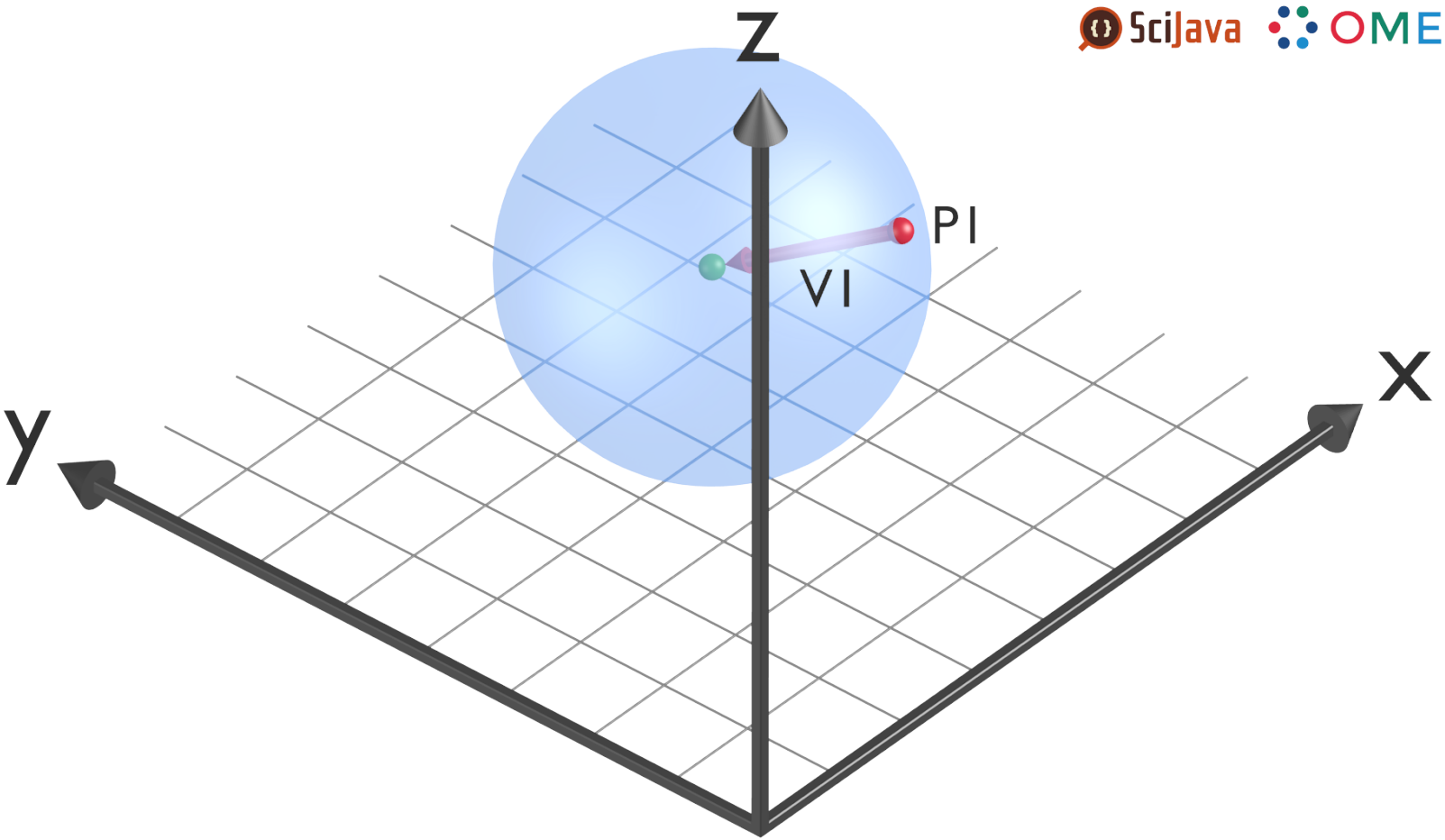
Describing a sphere: centre and radius



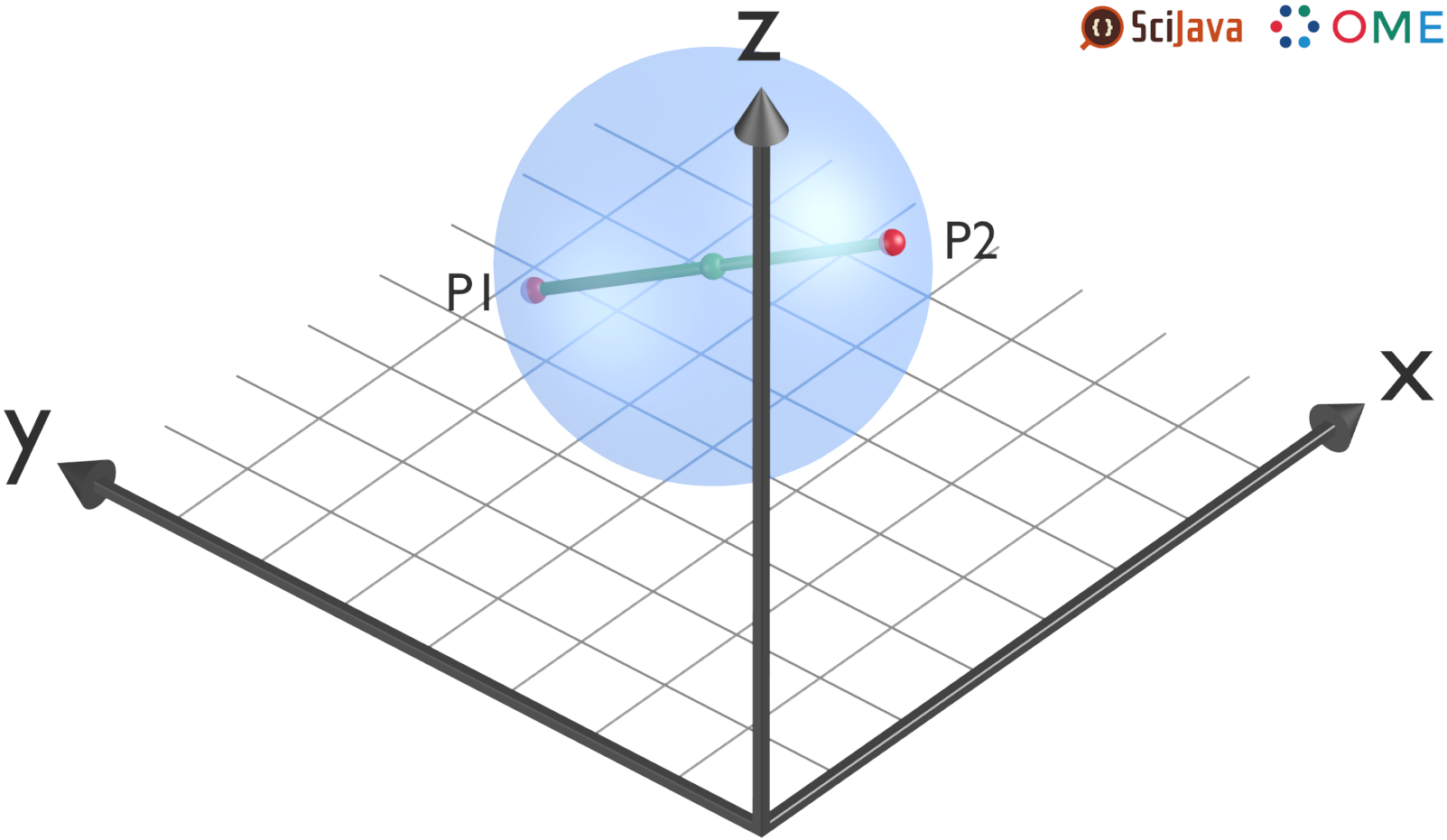
Describing a sphere: centre and radius



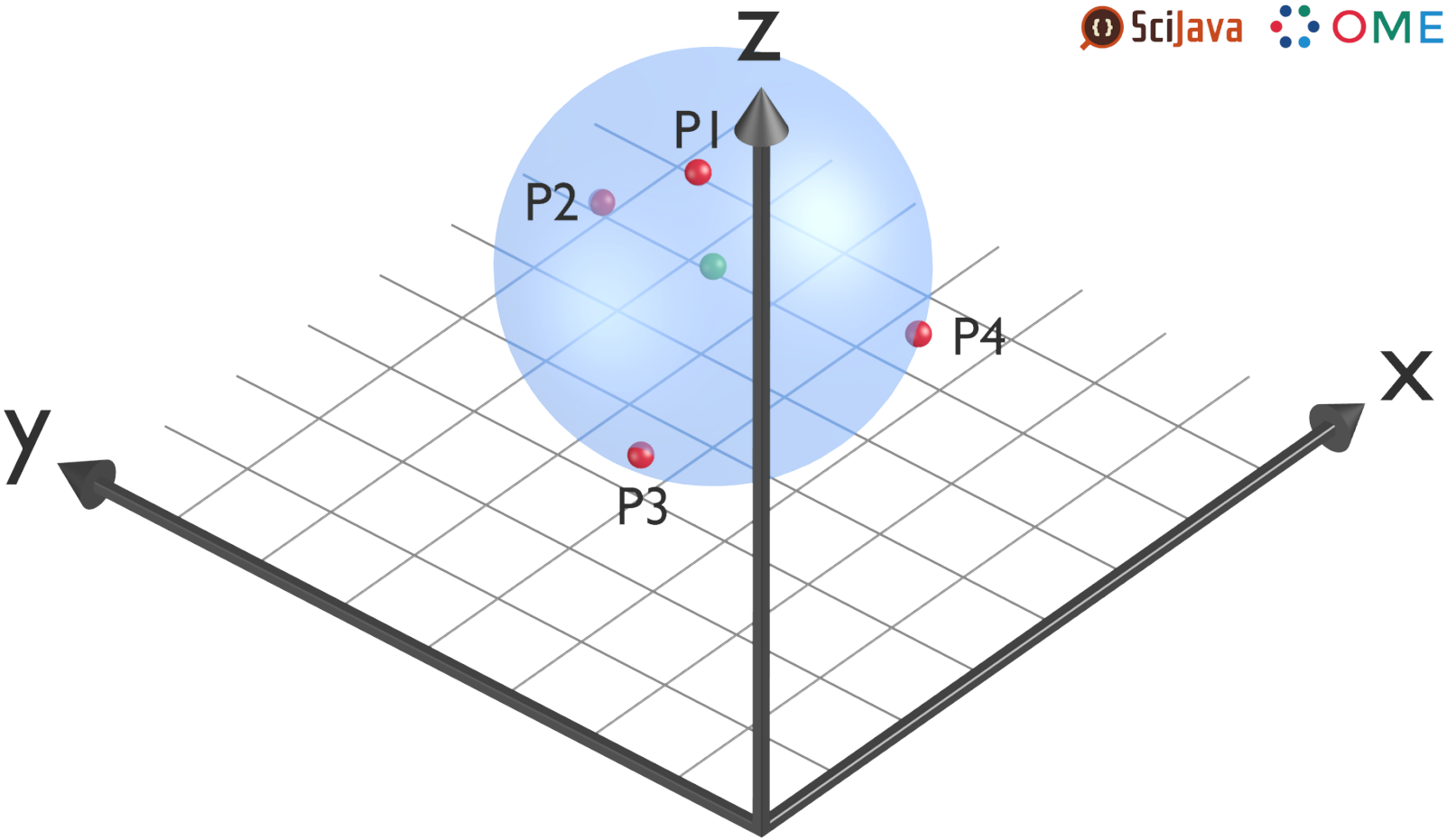
Describing a sphere: centre and radius



Describing a sphere: diameter



Describing a sphere: surface



A region of interest

- Shape
  - 3D geometric form
  - 2D shapes are described by a 1 pixel thick 3D shape
  - nD values or range
- Representation
  - How the shape is described
  - A shape may have one or more representations
  - One representation is the default or canonical representation for each shape
- Serialisation
  - A ROI is fully described by a shape ID, representation ID and the representation data (points, vectors, etc.)
  - Could be packed binary, text, XML, etc.

Shape types: 3D primitives

- 3D geometric forms (without volume)
  - Point
  - Points
  - Line
  - Lines
  - Polyline
  - Polygon
  - PolylineSpline
  - PolygonSpline
  - Arc
- 3D geometric forms (with volume)
  - Cuboid
  - Ellipsoid
  - Cylinder
  - Mesh
- User-definable 3D forms
  - Custom
- 3D pixel data
  - BitMask
  - GreyMask
- 3D transforms and operations
  - AffineTransform
  - AbstractTransform
  - Bitwise
- 3D Annotations
  - Text
  - Scale
  - Grid

Shape types: nD primitives

- nD constraints
  - Value
  - Values
  - Range
- nD transforms and operations
  - ExtrudeDim
  - CombineDim
- nD Grouping
  - Set
  - Group
- nD Metadata
  - Property

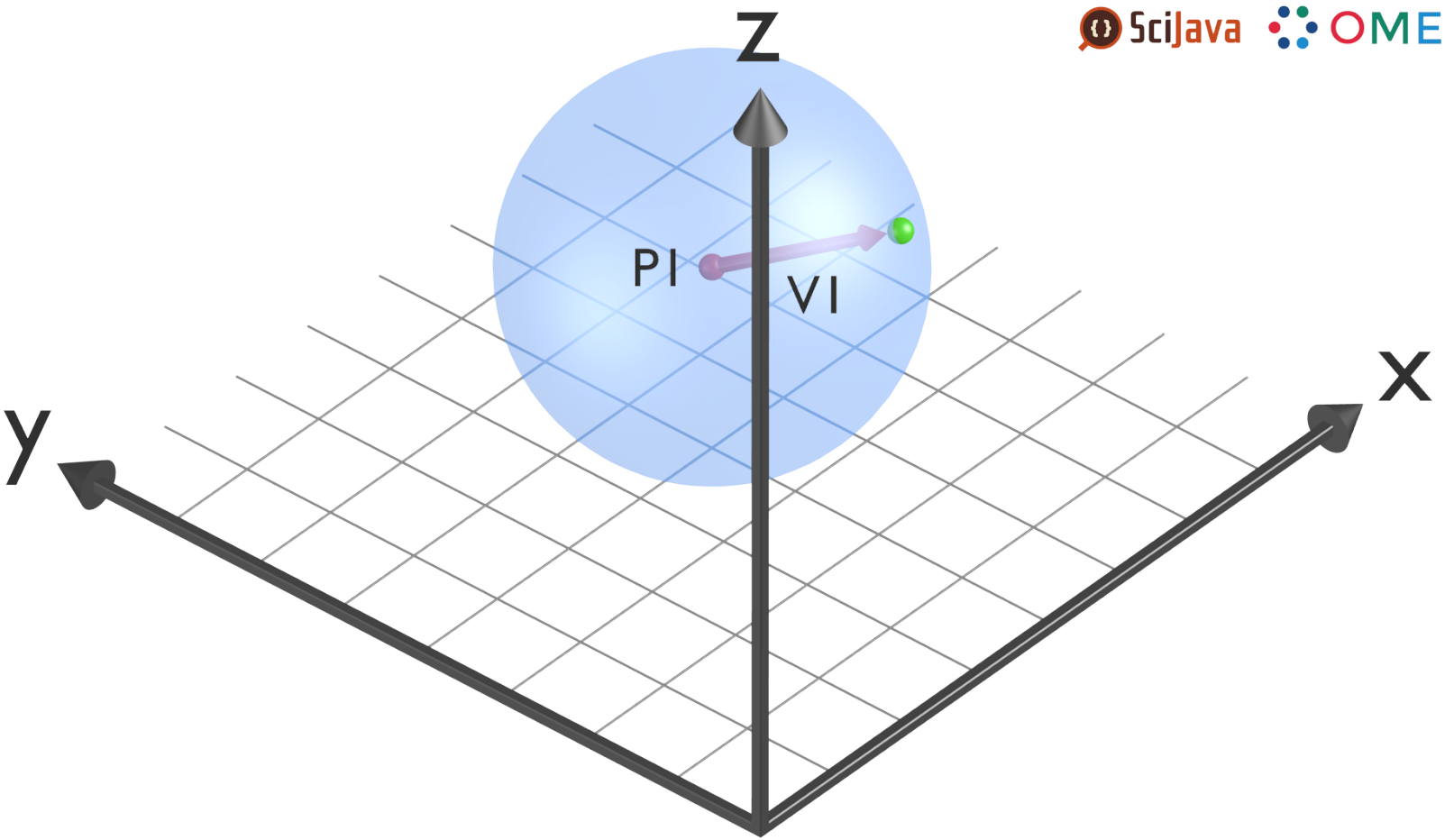
Representations: Ellipsoid

Representation	*Dim*	*In*	*Out*	*Canonical*
RSphere0	3D	true	true	false
RSphere1	3D	true	true	false
RSphere2	3D	true	true	false
RSphere3	3D	true	true	false
RSphere4	3D	true	true	false
RSphere5	3D	true	true	false
RSphere6	3D	true	true	false
RAlignedHalfAxes	3D	true	true	false
RHalfAxes	3D	true	true	true

Representation detail: Ellipsoid

Representation	*Dims*	*Seq*	*Name*	*Type*	*Description*
RSphere0	3D	0	P1	Vertex3D	Centre point
		1	P2	Vertex3D	Surface point
RSphere3	3D	0	P1	Vertex3D	Centre point
		1	V1	Vector3D	Radius
RSphere4	3D	0	P1	Vertex3D	Point on surface
		1	V1	Vector3D	Vector to centre
RSphere5	3D	0	P1	Vertex3D[2]	Two surface points
RSphere6	3D	0	P1	Vertex3D[4]	Four surface points
RAlignedHalfAxes	3D	0	P1	Vertex3D	Centre point
		1	V1	Vector3D	Half axes (x,y,z)
RHalfAxes	3D	0	P1	Vertex3D	Centre point
		1	V1	Vector3D	Half axes (xyz)
		2	V2	Vector2D	Half axes (xy)
		3	V3	Vector1D	Half axes (x)

Shape serialisation example: sphere centre and radius

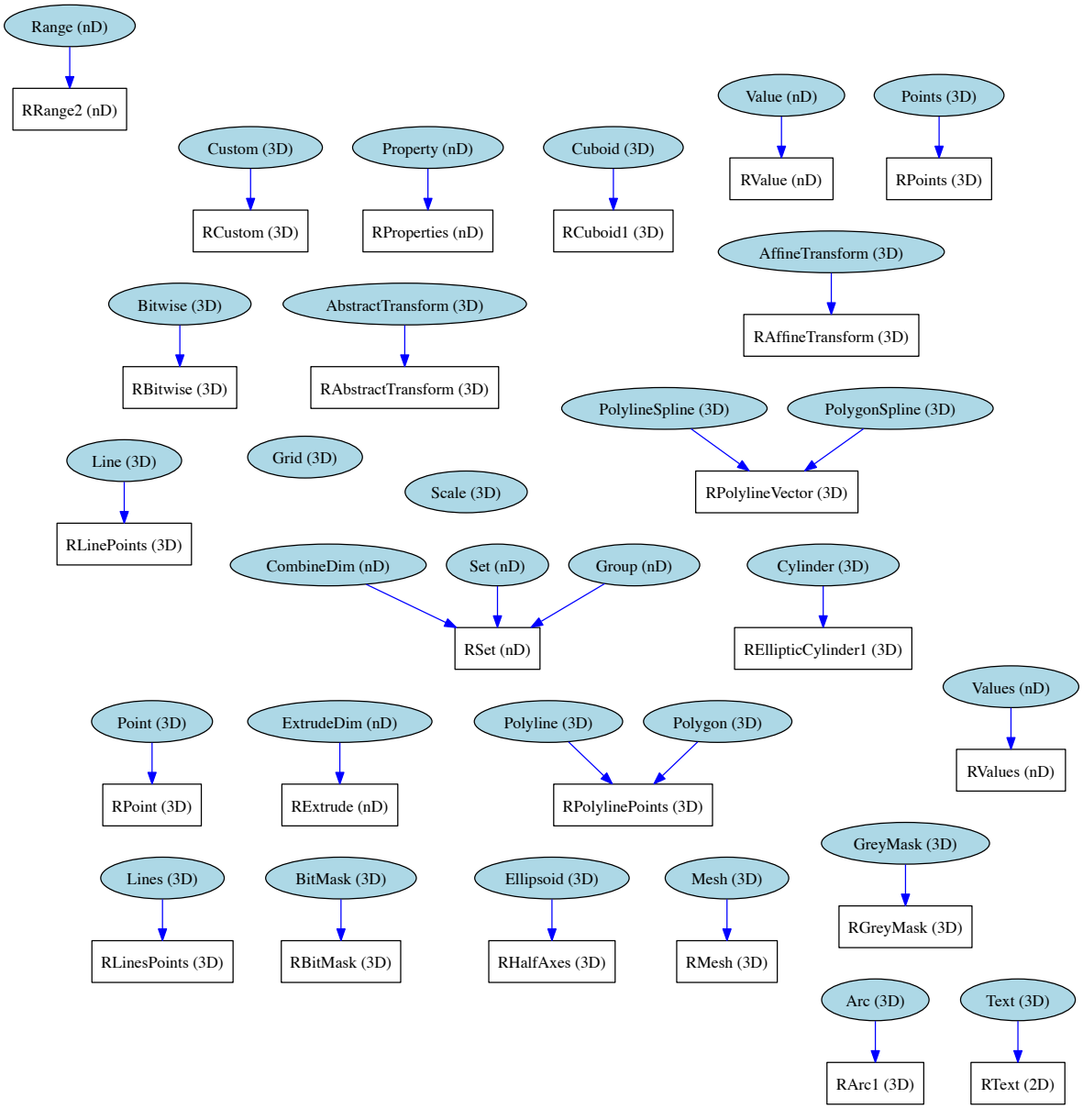


Shape serialisation example: sphere centre and radius

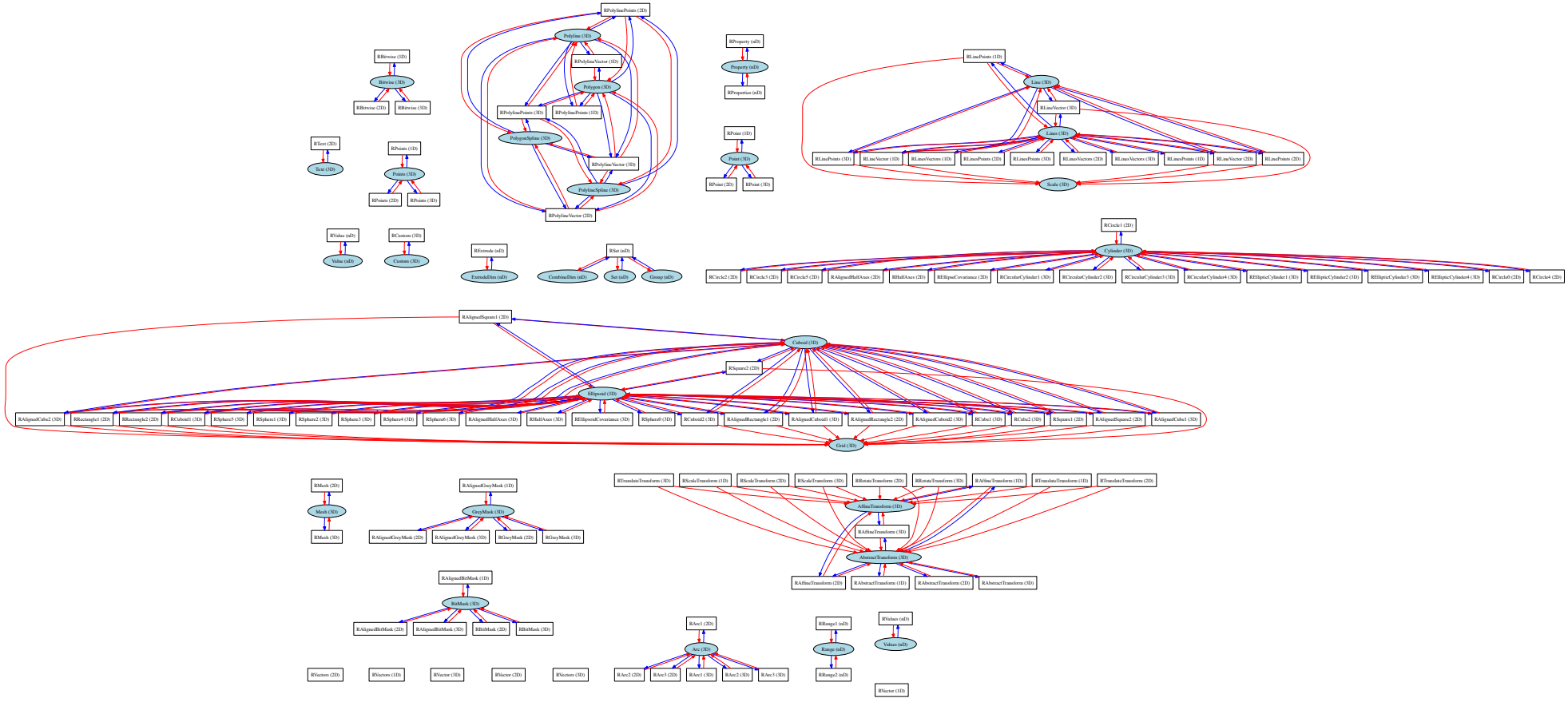
Name	*Type*	*Fundamental*	*Value*	*Description*
SID	ShapeID	uint16	11	Ellipsoid
RID	RepID	uint16	40	RSphere3
P1	Vertex3D	double	16.0	x
		double	16.0	y
		double	8.0	z
V1	Vector3D	double	1.716 53	x
		double	9.285 85	y
		double	11.397 16	z

Total size: 68 bytes.

Shapes and canonical representations



Shapes and all representations

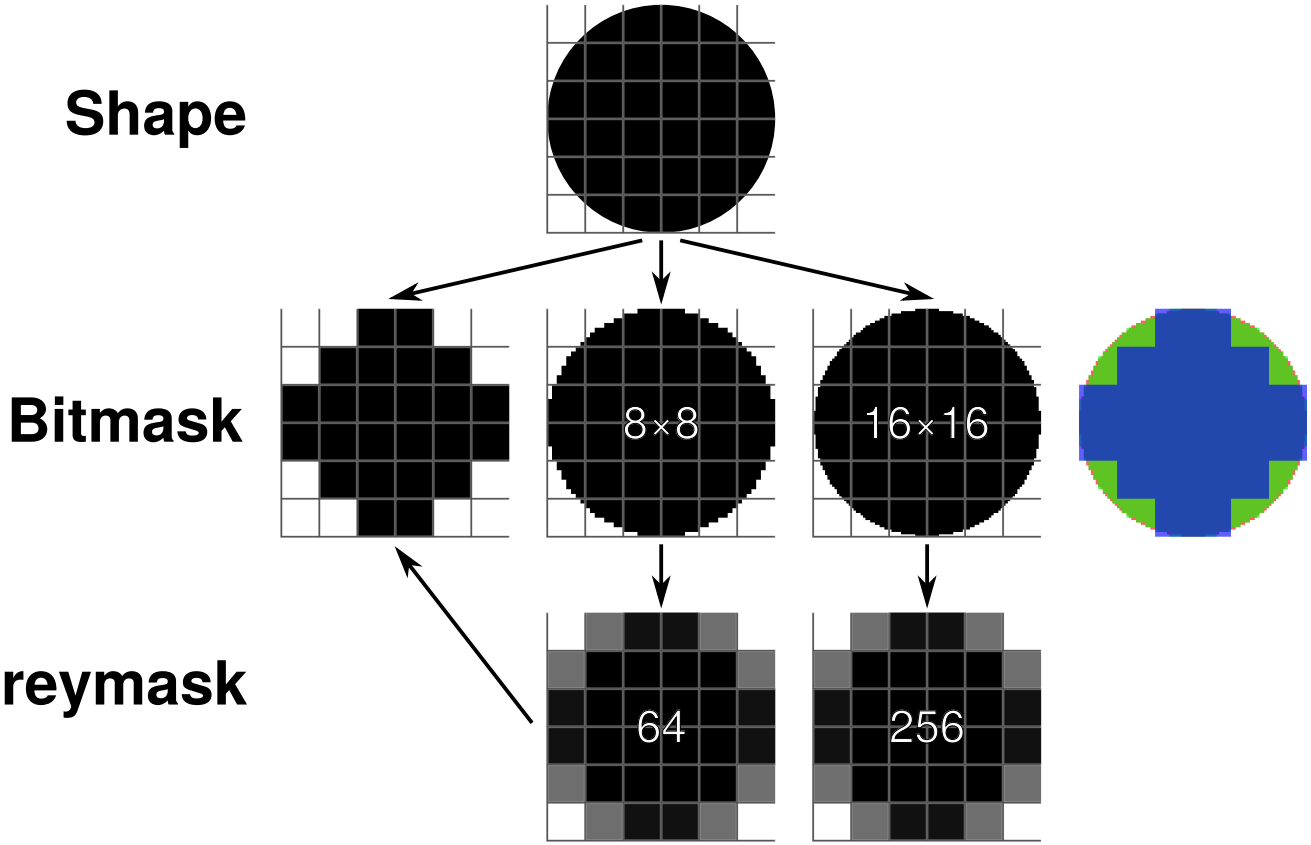


Shapes and transformations can stack

```
AffineTransform {
    Transform1
    Group {
        AffineTransform {
            Transform2
            Set {
                Shape1,
                Shape2
                AffineTransform {
                    Transform3
                    Shape3
                }
            }
        }
    }
}
```

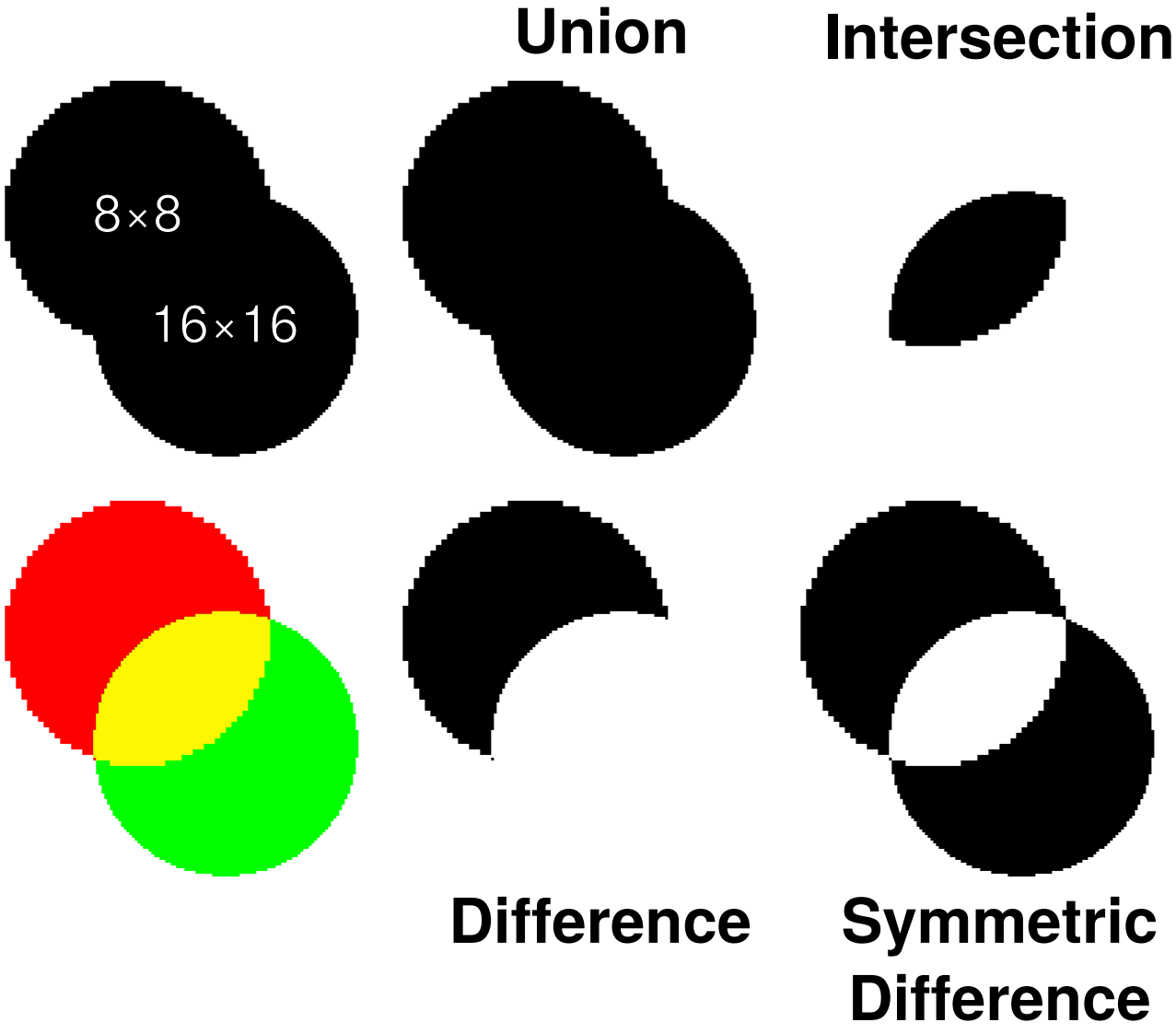
Bitmasks and greymasks

- Conversion of shape to bitmasks and greymasks



- Optimal storage for bitmasks?
- Alignment of masks with the image pixel grid?

Set operations on bitmasks



The resulting masks can be converted to a lower resolution greymask or bitmask.

Drawing

- Needs to be toolkit-independent
- All shapes draw using their canonical representation; one codepath for each shape type.
- Shapes reduce to:
  - Bitmask
  - Greymask
  - Mesh
- Viewers can all view masks or meshes in 2D or 3D
- OpenGL viewers can render meshes in 2D or 3D
- jHotDraw can render vectors where possible; unsupported complex types can be rendered in terms of simpler primitives
- There may be potentical loss of precision when converting; these forms are for visualisation only, not analysis.

Editing

- Edit in terms of the underlying shape representation
- Use the canonical representation where the original representation is not usable
- Edit in pixel, physical or other coordinate system
- Editing nodes and constraints specified by representation
- Possible to view the ROI as a treeview of nested shapes, and edit the properties of nodes in the view

Outstanding questions

- Grouping: what is a “ROI” cf. “Shape” or group/set of shapes? What is the boundary between a shape and a ROI?
- Rounded rectangles. Support as primitive or compound shape?
  - Danger of infringing *registered design No. 0000181607-0001*!
- Efficient mask storage: labellings, etc. Logic behind the different mechanisms? Convenience?
- Shape properties: what is currently in the different models?
- State machine properties for evaluating ROIs
- Dimension conventions: are shapes present in all absent/unspecified dimensions?

Needed work

- Agree on list of shape primitives
- Agree on list of representation primitives
- Agree on most appropriate canonical shape representations
- For each shape type, specify:
  - Mask conversion rules
  - Measurements
  - Editing rules
  - Drawing behaviour (greymap, jHotDraw, OpenGL etc.)
- Write code!
- Integrate and test code with programs using the ROI model

Acknowledgements

- OME, Dundee
  - Jason Swedlow
  - Chris Allan
  - Jean-Marie Burel
  - Will Moore
  - Andrew Patterson
- OME, Edinburgh / Harvard Medical School
  - Sébastien Besson