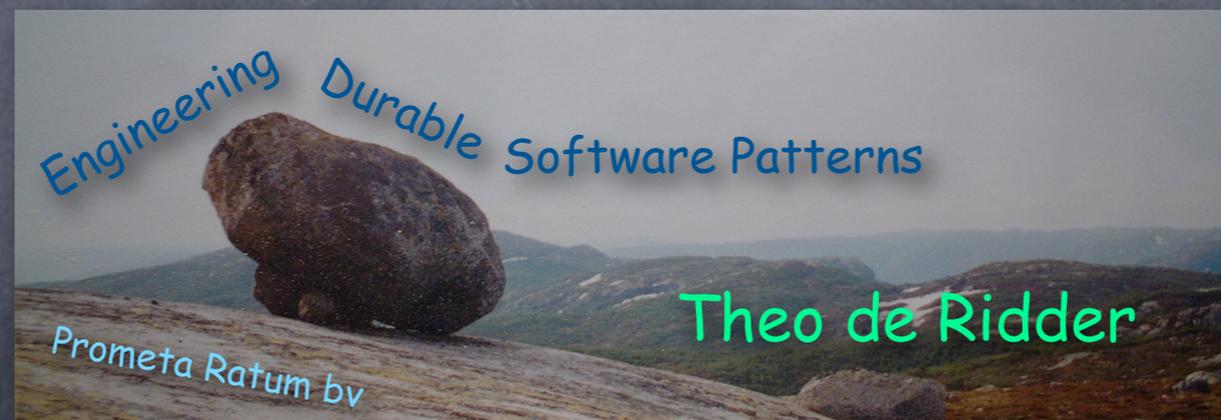


Visualizing behavior of ambient sensor networks

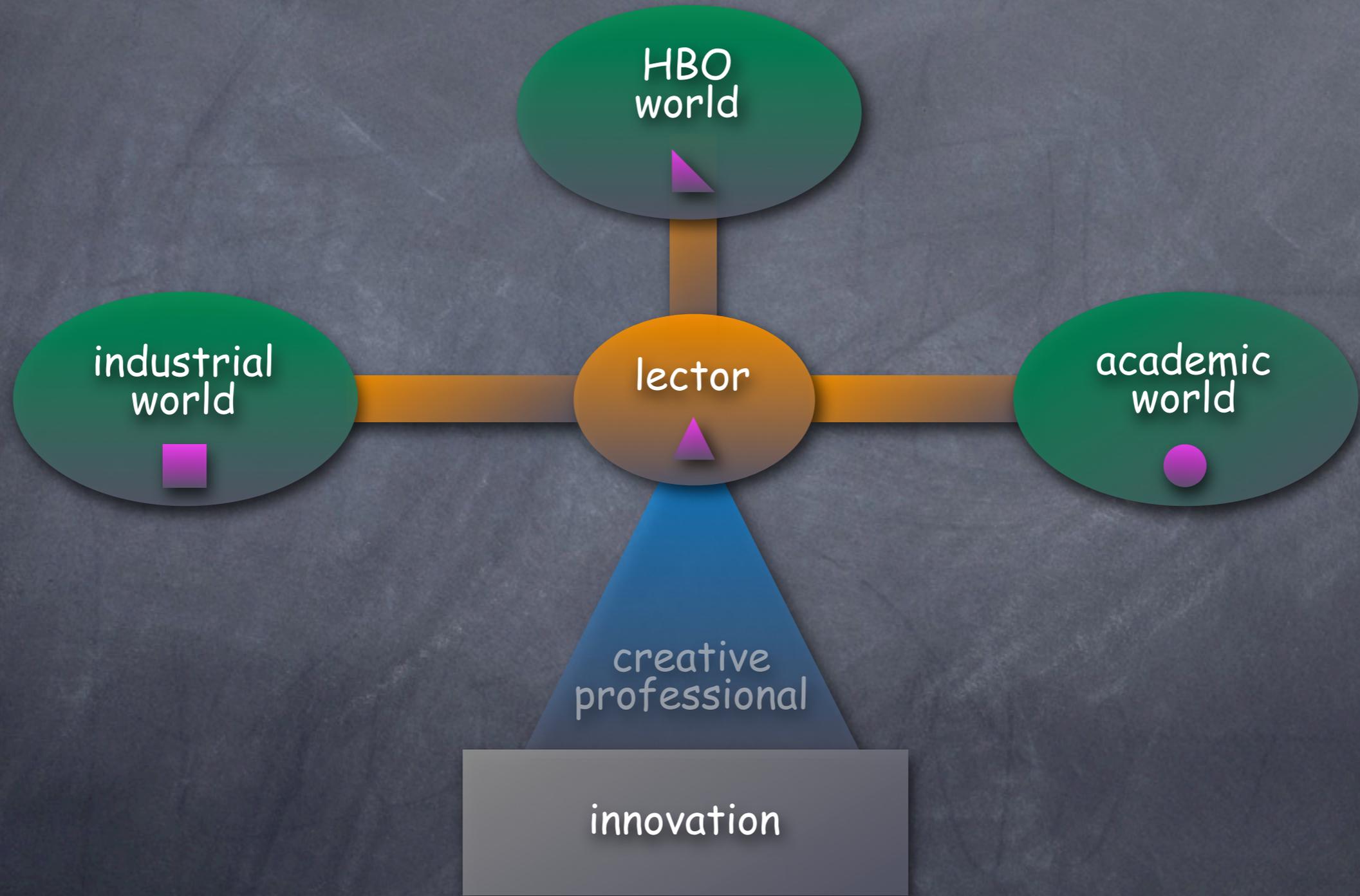


Europython 2006
Genève

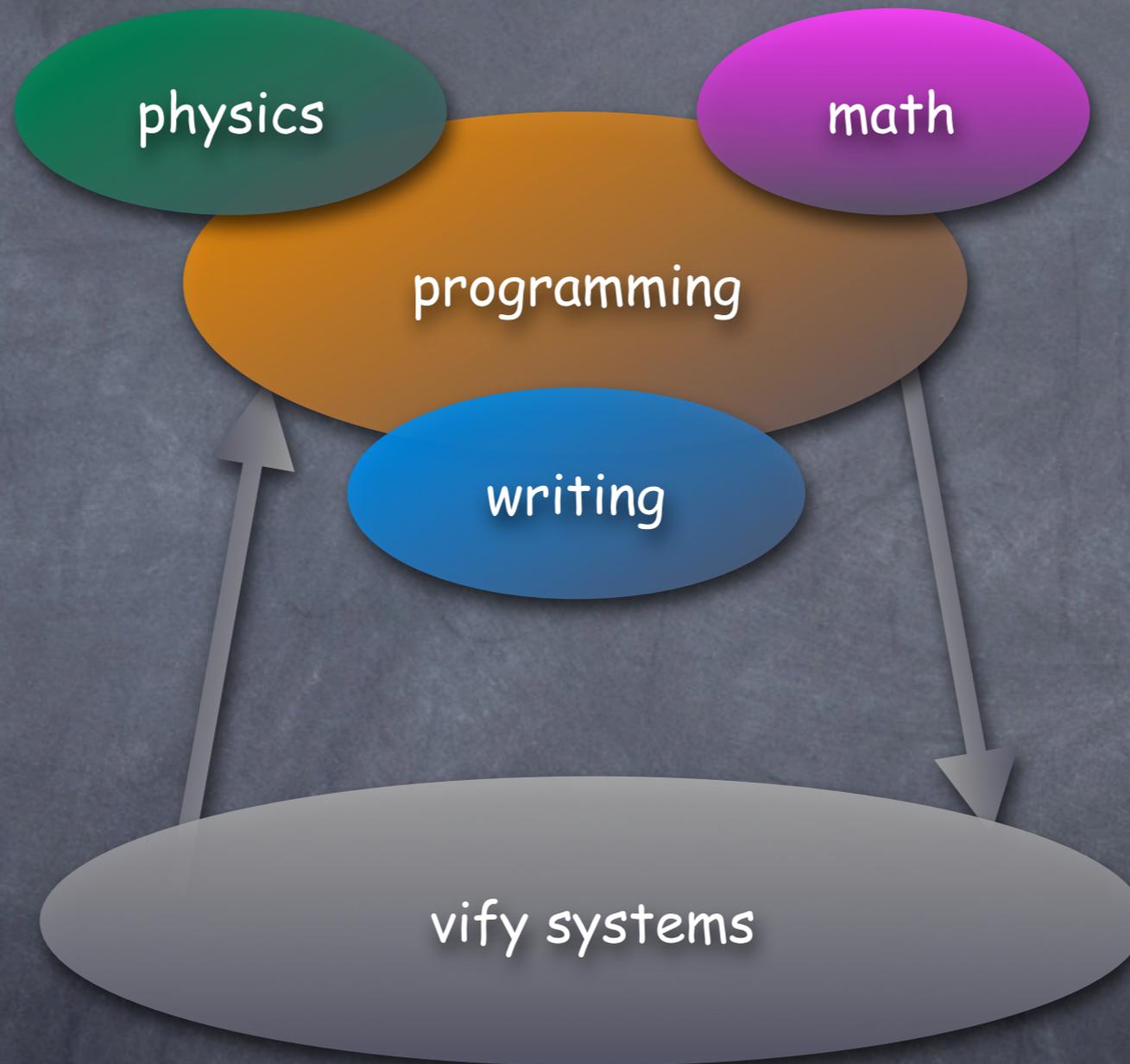
Agenda

- **Personal Position**
- DevLab as Research Context
- Large scale ad-hoc sensor networks
- Ambient Visualisation
- Moving Clusters
- Python as BisonKit

Professional

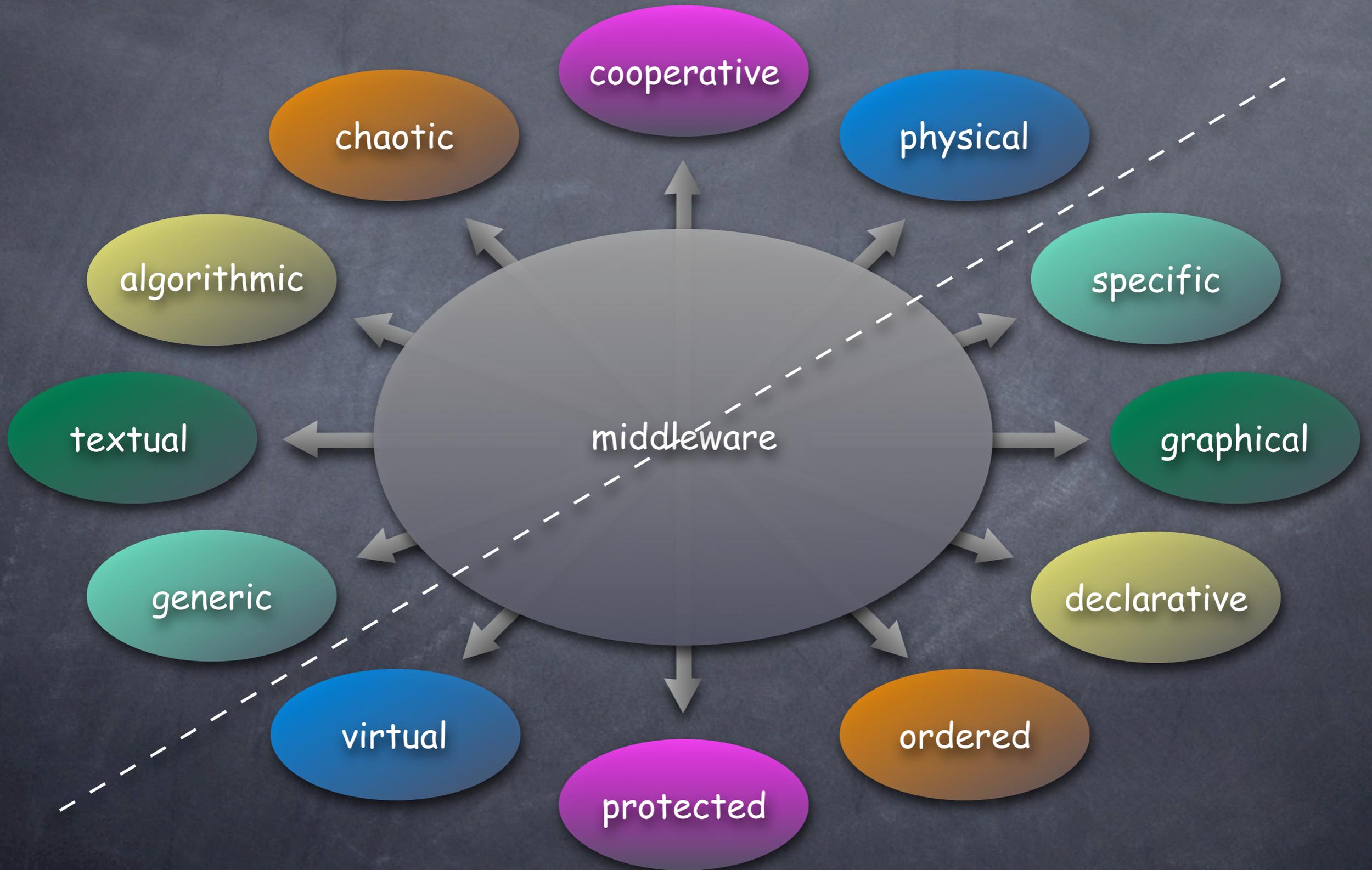


Artistic

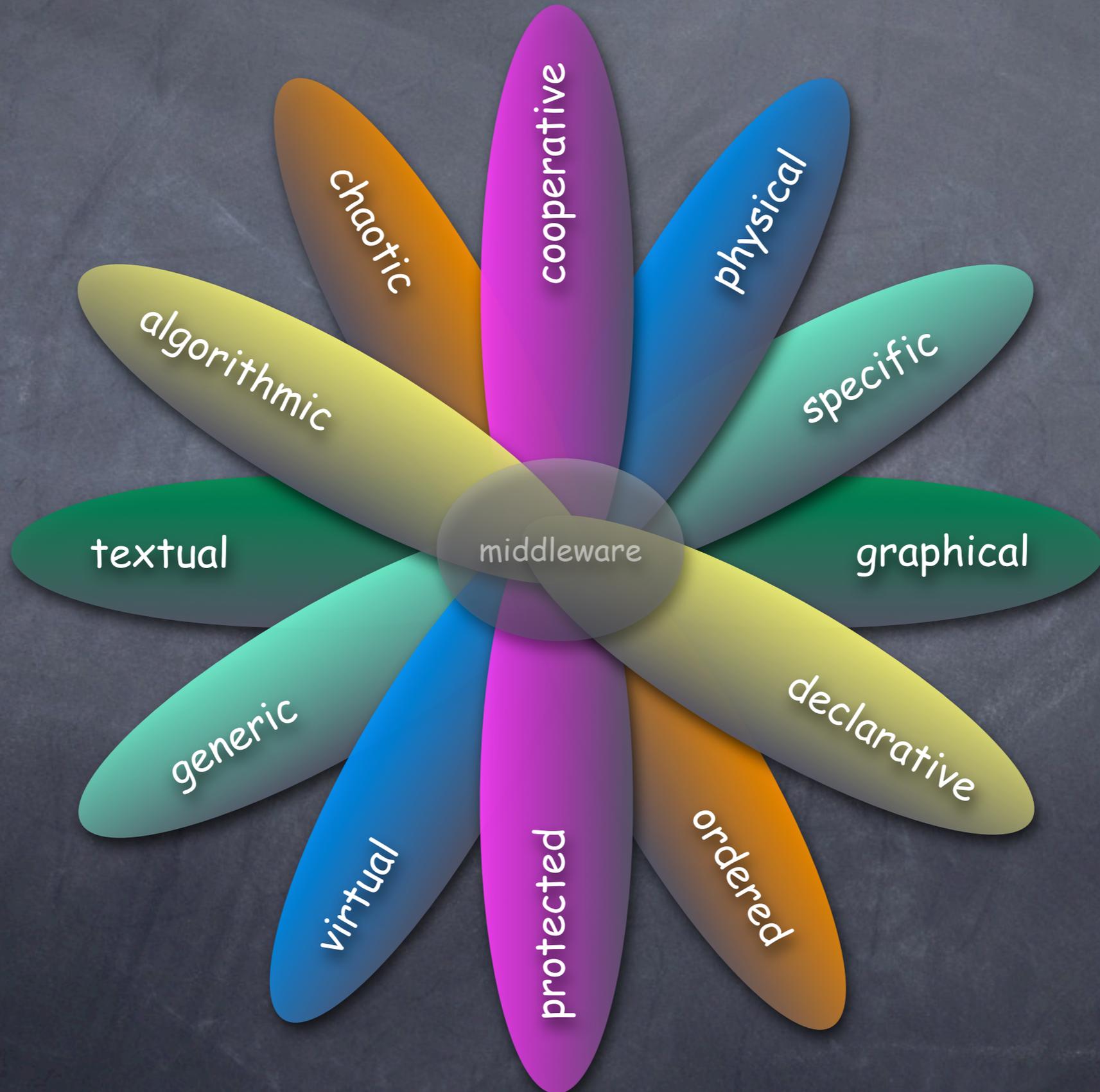


"Elegance as multi-disciplinary glue"

Conceptual



Conceptual



Pythonic

unique



disruptive



familiar

traditional

Pythonic

unique



disruptive

Python

familiar

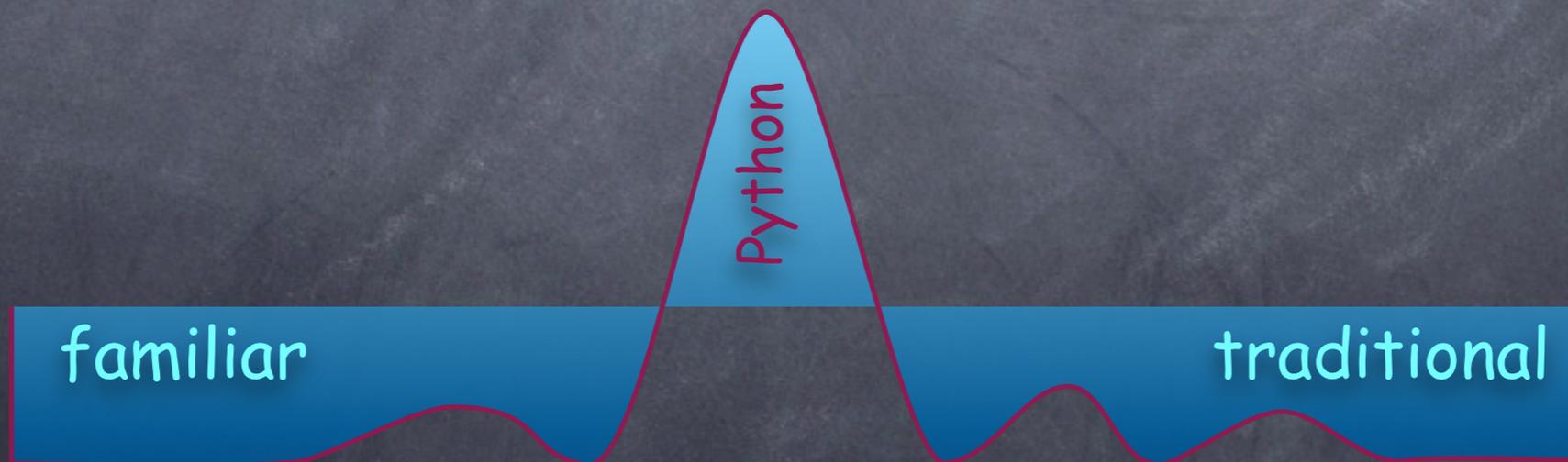
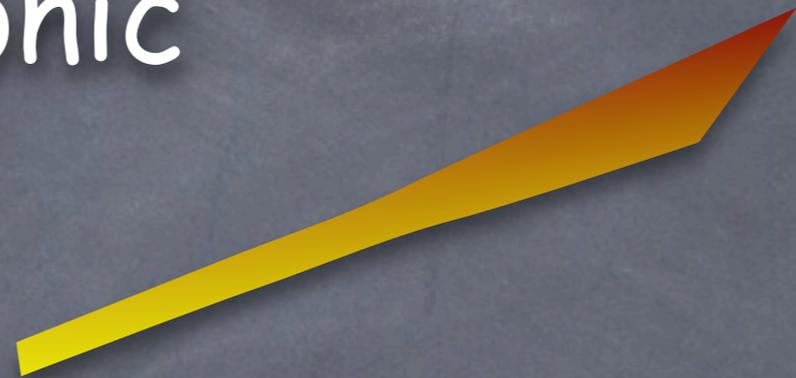
traditional

Pythonic

unique



disruptive



Agenda

- Personal Position
- **DevLab as Research Context**
- Large scale ad-hoc sensor networks
- Ambient Visualisation
- Moving Clusters
- Python as BisonKit

DevLab

- **DevLab (Development Laboratories)**
 - cooperation of 12 SME high-tech companies
 - started 1 februar 2005
- **goals**
 - business creation by explorative research
 - early adoption of emerging technologies
- **characteristics**
 - active partnership of SME companies, polytechnics, universities and research institutes
 - SME companies in control (coop member => 1% of wages)
 - mutual trust and respect

two starting 'products'

- **Atalanta**

mechatronic butterfly as technology integrator

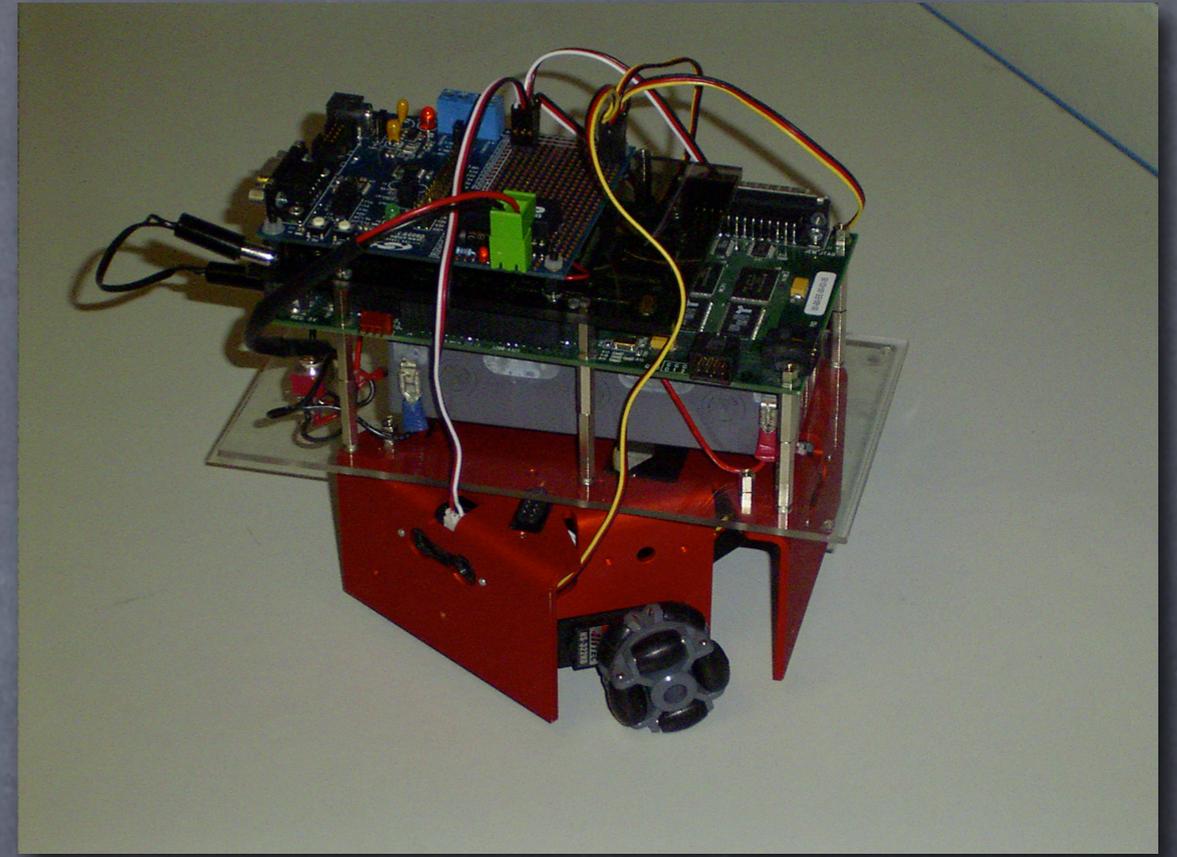
- energy harvesting
- observing, reaction, flying
- self-configuring swarms
- realistic size and weight

- **Dutch Clay**

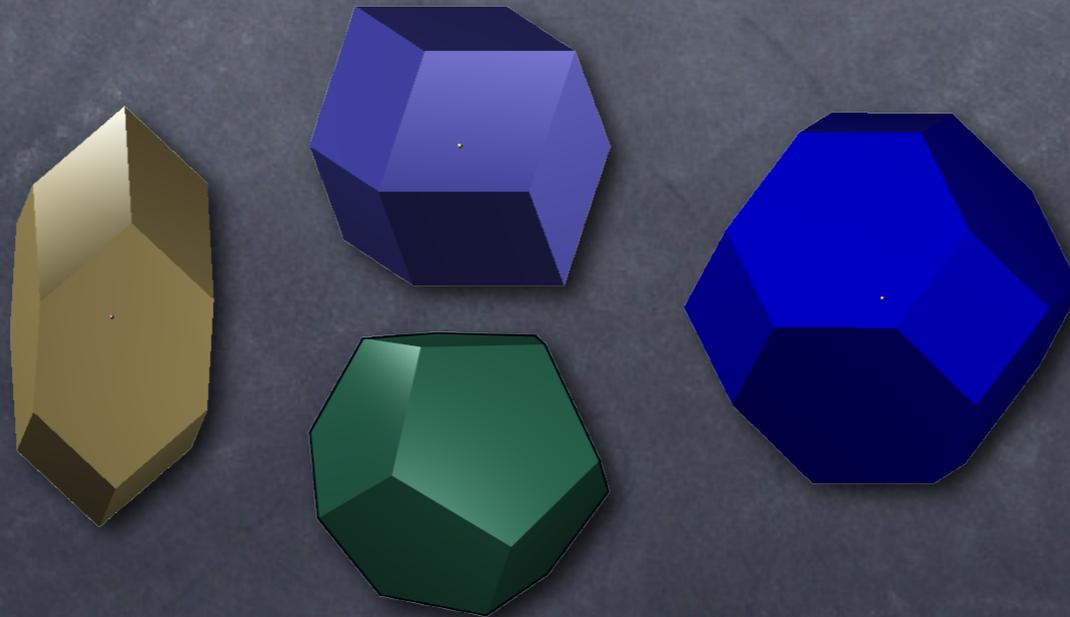
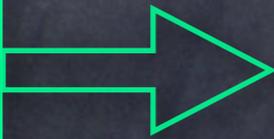
haptic "smart" material

- immediate self-consciousness of shape
- ad-hoc clusters of identical grains
- lighting, elastic, transparent
- from 100 pingpongballs towards 100000 grains

virtual & physical



Blender
+
Python

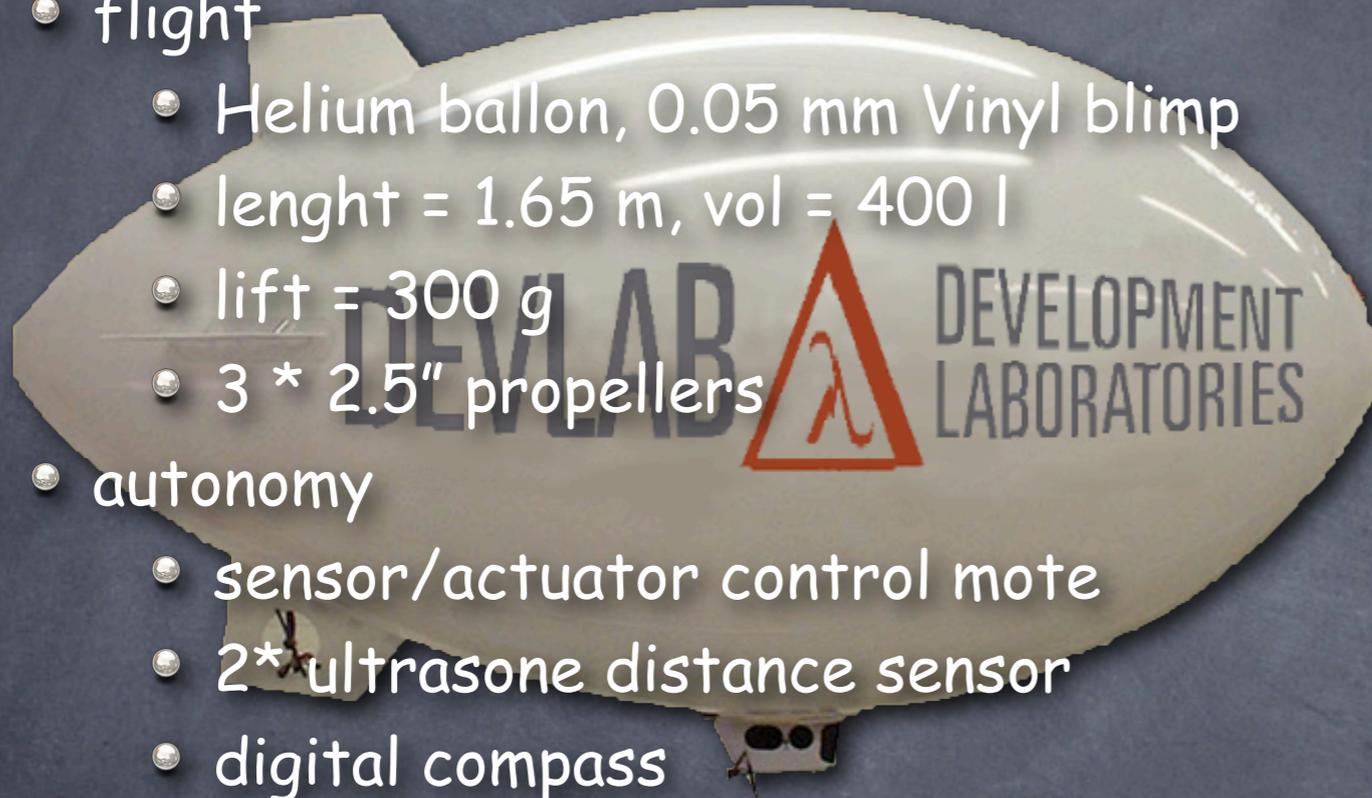


embedded
Java
hardware

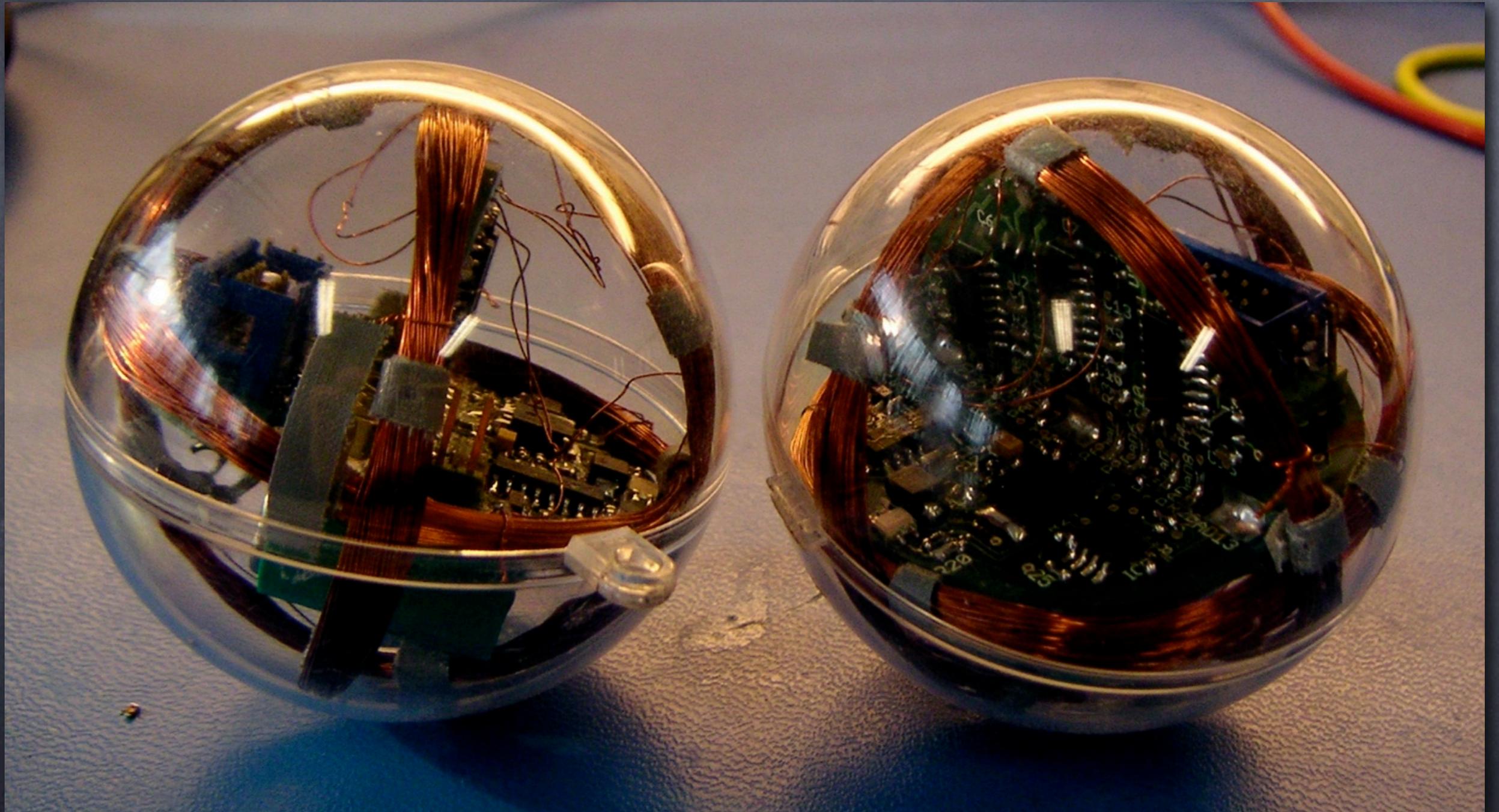


testing autonomous indoor flight

- flight
 - Helium ballon, 0.05 mm Vinyl blimp
 - lenght = 1.65 m, vol = 400 l
 - lift = 300 g
 - 3 * 2.5" propellers
- autonomy
 - sensor/actuator control mote
 - 2* ultrasonic distance sensor
 - digital compass
 - infrared sensor
 - radio receiver
 -



first physical claymotes



Agenda

- Personal Position
- DevLab as Research Context
- **Large scale ad-hoc sensor networks**
- Ambient Visualisation
- Moving Clusters
- Python as BisonKit

MyriaNed

- **characteristics**

- large scale, ad-hoc, low-power
 - 10K motes, $\sim \text{cm}^2$
 - 50-100 m, 5-10 years
- heterogenous, Lan & Wan
- self-organising
- industrial quality

- **aspects**

- zero maintenance
- reliability
- flexibility
- adaptivity
- observability
- security



biological inspired systems

- **self-organising crowds of unstable components**
 - survival strategy: insensitivity to threads
 - robustness
 - adaptivity
 - independence of central control
 - interaction patterns
 - diffusion, replication
 - reaction-diffusion
 - chemotaxis, stigmergy
- **flying insects**
 - flapping for flight requires high-tech engineering
 - exploiting resonance is a must
 - energy balance is delicate

Minimal protocols

- **NewsCast**
 - gossip
 - ad-hoc, broadcast, multi-hop, no ack
- **TOTA (Tuples On The Air)**
 - triple $T = (C, P, M)$
 - Content
 - Propagation Rule
 - Maintenance Rule
 - Flocking with Fields
 - Routing with Pheromones
- **GlassBoard**
 - distributed shared execution space
 - publish-subscribe of footprints
 - reversing client-server roles
 - server
 - poor, obedient, transparent
 - client
 - rich, powerful, self-protecting

Agenda

- Personal Position
- DevLab as Research Context
- Large scale ad-hoc sensor networks
- **Ambient Visualisation**
- Moving Clusters
- Python as BisonKit

Purposes

- Visualiser of state/context correlations
- Navigator for connected "villages"
- Zoomlens for debugging and testing
- Simulator for application scenario's
- Marketing instrument

Architecture



Interactive Demo

The image shows a screenshot of the Blender 2.79 software interface. The main 3D viewport displays a scene with a butterfly (wing-R) in the foreground, a cluster of small spheres, and several larger spheres with colored patches. A red triangle is visible in the background. The interface includes a top menu bar with options like File, Add, Timeline, Game, Render, and Help. Below the menu bar, there are tabs for SR:1-Animation, SCE:Scene, and a browser for www.blender.org. The left sidebar shows a list of objects in the scene, including Ball-A through Ball-Y, Blimp, Caspar, Cube-n, Cube-z, Cylinder, Cylinder.001, Delta, Floor, Jorm, Lambda, Nooduitg, Ring-1 through Ring-3, Route, SphereCluster, Walls, Wim, and kop-theo. The right sidebar shows a graph editor with a sine wave plot for rotation values (RotX, RotY, RotZ) over time. The bottom of the interface features a Properties panel with sections for Preview, Texture, Colors, Image, and Anim and Movie. The Preview section shows a butterfly image. The Texture section shows a texture named TE:Tex.020. The Image section shows a texture named /atalanta2.jpg. The Anim and Movie section shows animation settings for the selected object.

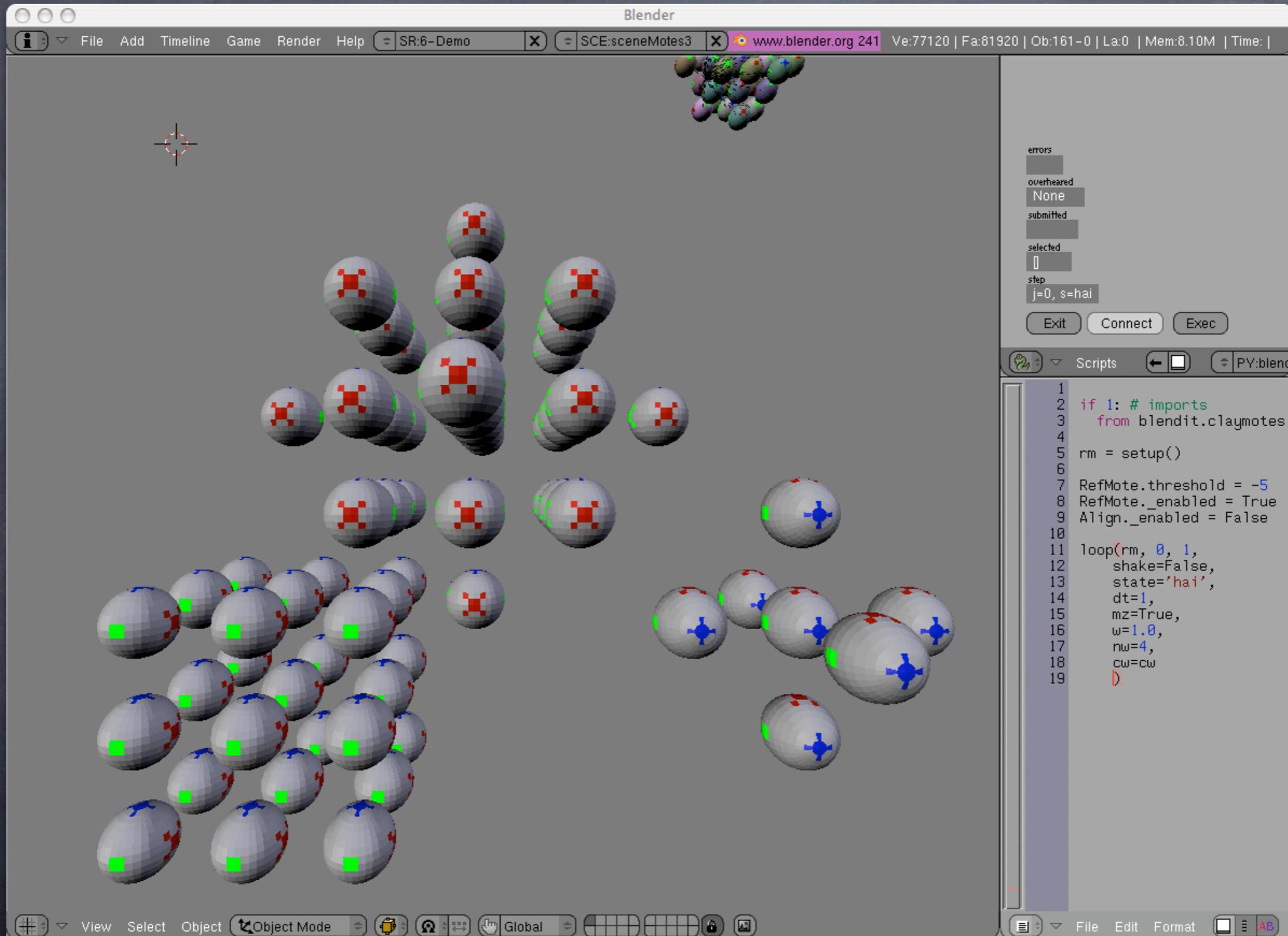
Rendered Animation



Agenda

- Personal Position
- DevLab as Research Context
- Large scale ad-hoc sensor networks
- Ambient Visualisation
- **Moving Clusters**
- Python as BisonKit

Moving Motes Demo



The image shows a Blender 2.49 interface. The main 3D viewport displays a scene with numerous small, semi-transparent spheres (motes) arranged in a cluster. Some spheres have red 'X' marks, some have green squares, and some have blue crosses. A small cluster of multi-colored spheres is visible at the top center. The interface includes a menu bar (File, Add, Timeline, Game, Render, Help), a status bar (SR:6-Demo, SCE:sceneMotes3, www.blender.org 241, Ve:77120 | Fa:81920 | Ob:161-0 | La:0 | Mem:8.10M | Time: |), and a bottom toolbar (View, Select, Object, Object Mode, Global). On the right side, there is a console window showing error messages and a Python script editor with the following code:

```
errors
overheared
None
submitted
selected
[]
step
j=0, s=hai
Exit Connect Exec

Scripts
PY:blenc
1
2 if 1: # imports
3     from blendit.claymotes.
4
5 rm = setup()
6
7 RefMote.threshold = -5
8 RefMote._enabled = True
9 Align._enabled = False
10
11 loop(rm, 0, 1,
12     shake=False,
13     state='hai',
14     dt=1,
15     mz=True,
16     w=1.0,
17     nw=4,
18     cw=cw
19 )
```

Agenda

- Personal Position
- DevLab as Research Context
- Large scale ad-hoc sensor networks
- Ambient Visualisation
- Moving Clusters
- **Python as BisonKit**

Mote & Space

```
if 1:      # imports
if 1:      # globals defs
class Map (object):
class Hole (object):
class Mote (Hole):
class Receiver (list):
class Protocol:
class Hai (Protocol):
class RefMote (Protocol):
class Align (Protocol):
class Space (object):
class Cluster (Space):
```

Space grids in Numpy

```
class Space (object):
    def __init__ (self, shape=(5,5,5), filler=Hole()):
    @property
    def filled (self):
    @property
    def center (self):
    def step (self, w=None):
        if w != None:
            self.ar * w
        Map('calc').u1(self.ar)
        s = [slice(None)]
        n = self.ndim
        for i in range(n):
            k1 = (i)*s + [slice(None, -1)] + (n-i-1)*s
            k2 = (i)*s + [slice(1,None)] + (n-i-1)*s
            self.facing = 2*i
            self[k1] << self[k2]
            self.facing = 2*i + 1
            self[k2] << self[k1]
```

Voting as local invariant

```
16 = class RefMote (Protocol):
17     _enabled = True
18     threshold = -15
19
20 = def reset (self):
21     self.ref = 0,0
22     self.agreed = []
23
24 = def __call__ (self, ref={}):
25     ref0 = (self.mnr, 0)
26     nr,age = max([ref0, self.ref] + ref.values())
27     if age > self.threshold:
28         self.ref = nr,age-1
29     else:
30         self.ref = ref0
31     dict(ref=self.ref) >> self
32 = self.agreed = [r for r in ref.items()
33                 if r[1][0] == self.ref[0]]
```

Visual wrappers in Blender

```
9  ⊞ class DupMesh (NMesh):
38 ⊞ class BallObject (Object):
88 ⊞ class Ball (DupMesh):
93 ⊞ class RoomObject (Object):
94   |     __slots__ = ['scene', 'space', 'scale', 'clusters']
95   |
96 ⊞   |     def init (self):
107 ⊞  |     def cluster (self, name, which='cube', shape=5):
133 ⊞  |     def update (self, *names, **kwargs):
146 ⊞  |     def shake (self, *names):
152 ⊞  |     def drag (self, name, pos=None):
162 ⊞  |     def fixate (self, name, mnr=None):
```


non-dogmatic paradigm mixing

concept	paradigm	technique	merits
(story)telling	imperative	module suite	higher readability
structuring	OO	class/method	less branching complexity
mapping	functional	iterators/ matrices	higher looping performance
representing	declarative	meta-programming	unified domain transparency





the end