INScore

AN ENVIRONMENT FOR THE DESIGN OF LIVE MUSIC SCORES

D. Fober, **Y. Orlarey,** S. Letz

Linux Audio Conference 2012 - April 12-15 @ CCRMA, Stanford University, CA, USA

Interactive Music Scores

Interactive Music Scores

- Alien Lands - Sandeep Bhagwati
  [Montréal - February 2011]
  Music performance in four movements for four spatially dispersed percussionists with interactive scores.

- Calder's Violin - Richard Hoadley
  [Cambridge - October 2011]
  Automatic music for violin and computer.

The Interlude Project

New Digital Paradigms for Exploration and Interaction of Expressive Movement with Music.

The Interlude Project

INScore

- Music score extension
- Graphic & time spaces relationship
- Performance representation
- Interaction

INScore

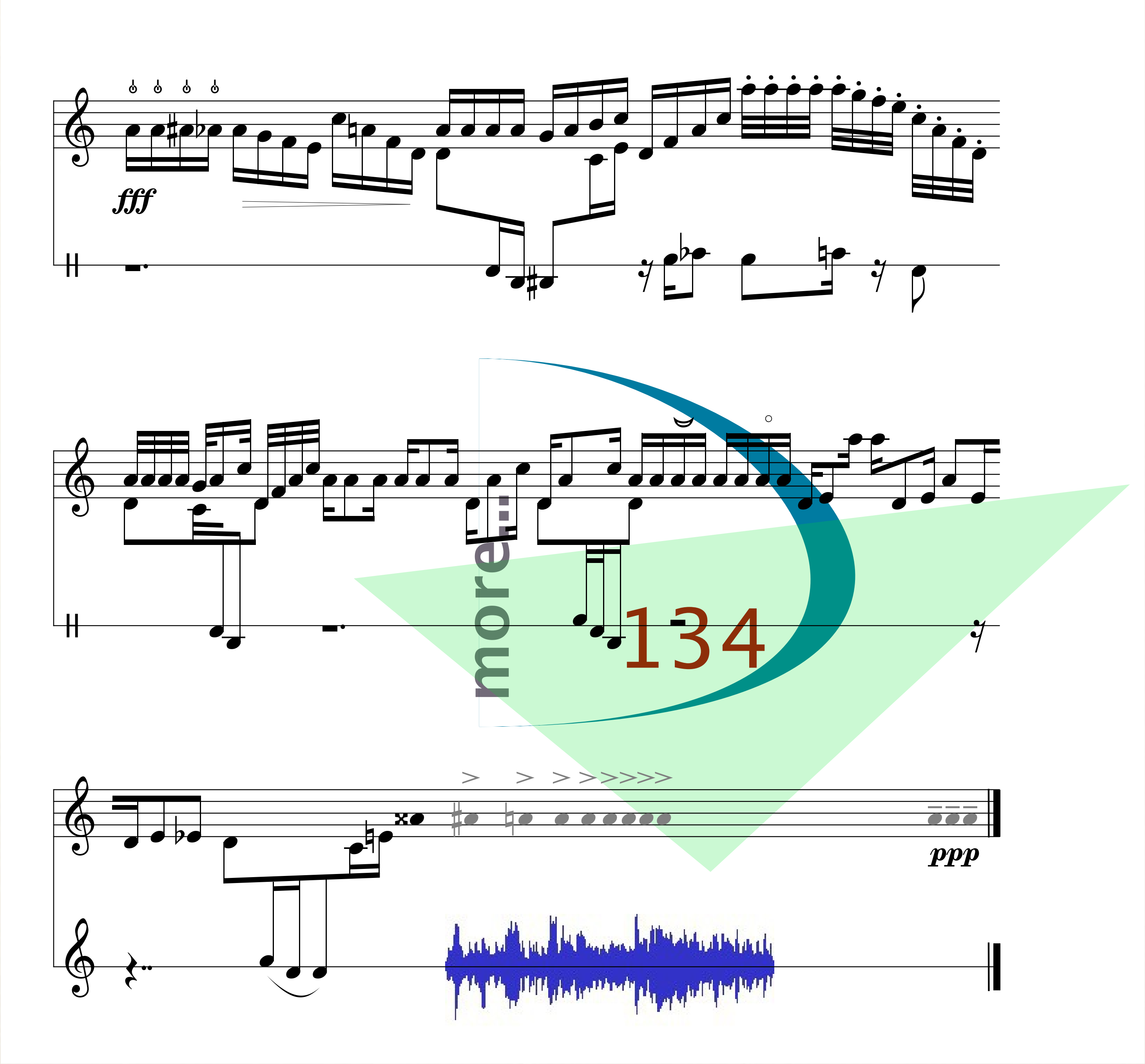
INScore supports

- Symbolic music notation [GMN, MusicXML]
- Textual elements
- Bitmaps [jpg, gif, tiff, png,...]
- Vectorial graphics (rectangles, ellipses, SVG,...)
- Video files
- Sound and gesture graphic representations

INScore is

- a standalone score viewer
- an open source C/C++ library
- multi-platform
- an Open Sound Control API

INScore



Relations between graphic and time space

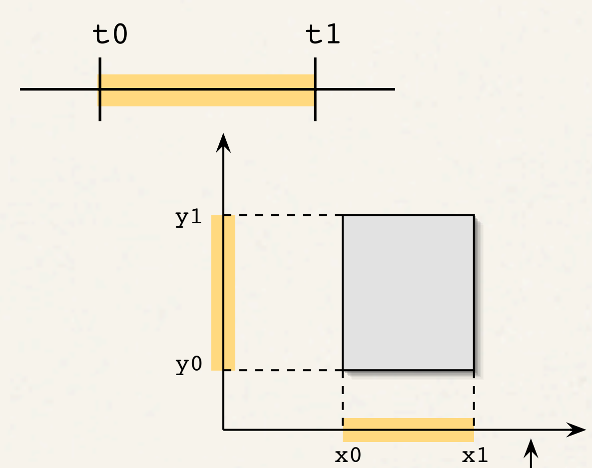
Hypothesis

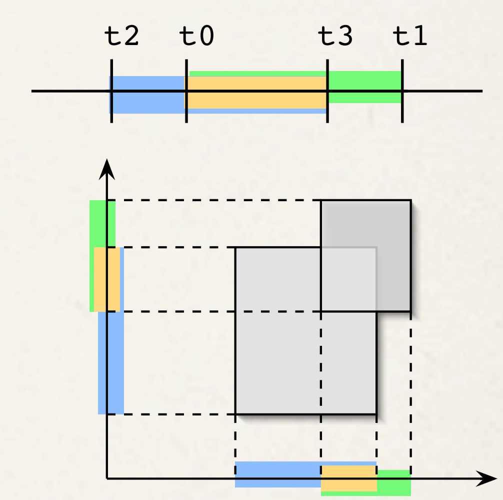
Approach the problem with segmentation and relations between segments

Relations between graphic and time space

Hypothesis

Approach the problem with segmentation and relations between segments

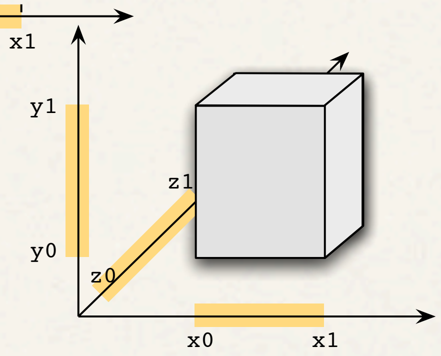




Segments

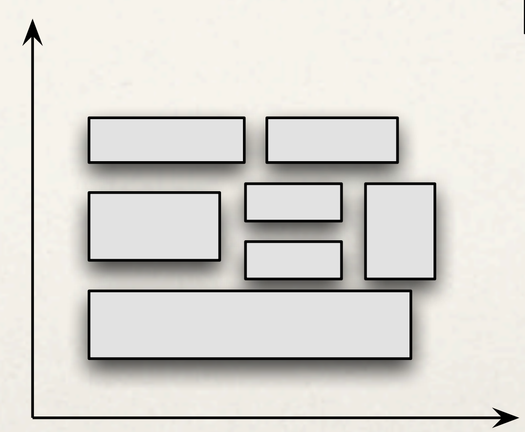
Defined as a list of intervals:

- generalizable to n dimensions
- intersection operation



Segmentation

A set of disjoined segments



Relations between graphic and time space

Hypothesis

Approach the problem with segmentation and relations between segments

Relations between graphic and time space

Hypothesis

Approach the problem with segmentation and relations between segments

Mapping

Relation between two segmentations:

- operations to query the mapping
- operations to compose mappings

Relations between graphic and time space

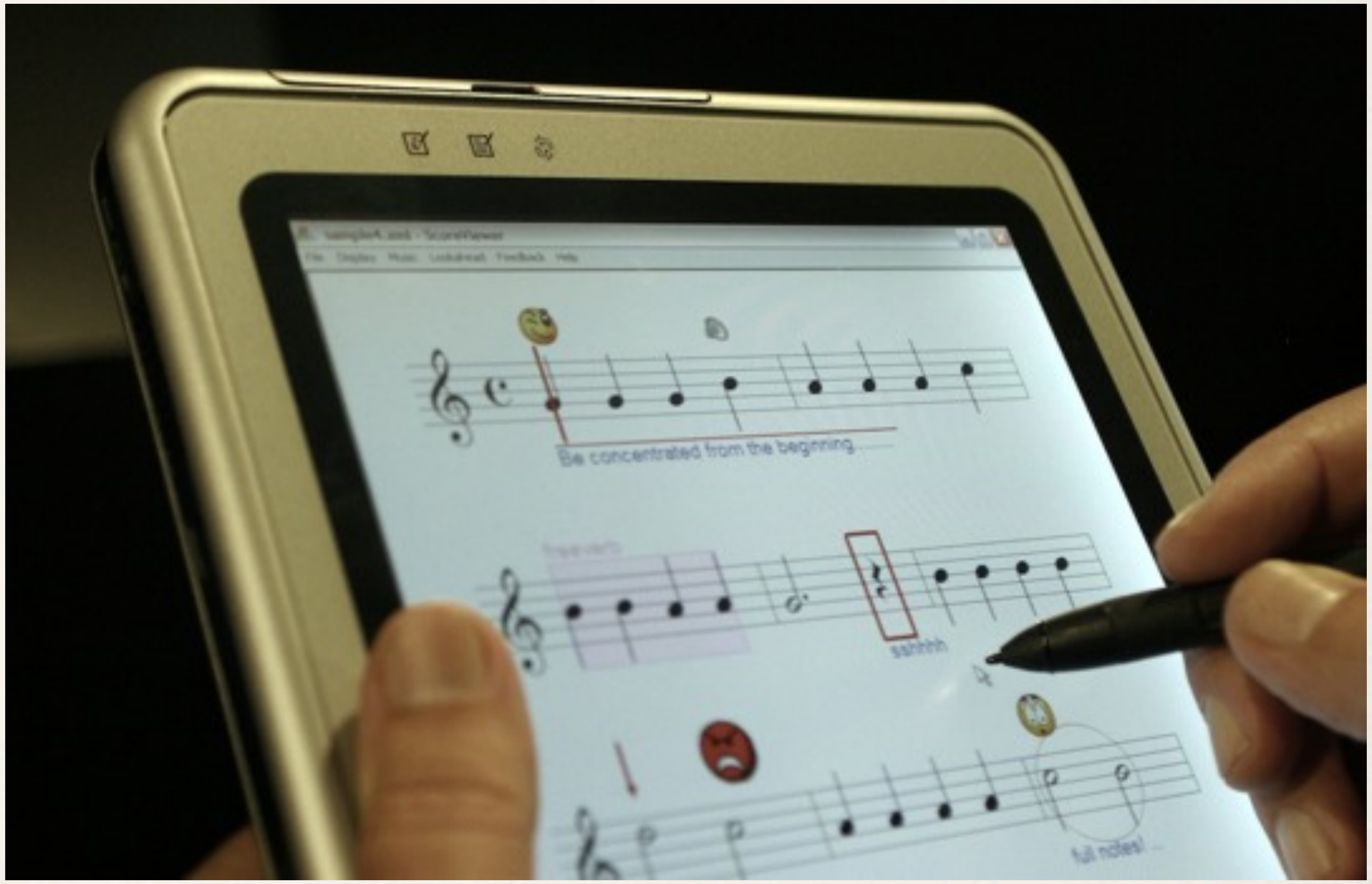
Segmentations and mappings for each component type:

type	segmentations and mappings required
text	*graphic* ↔ **text** ↔ **relative time**
score	*graphic*↔ *wrapped relative time* ↔ *relative time*
image	*graphic* ↔ **pixel** ↔ **relative time**
vect. graphics	**vectorial**↔ **relative time**
signal	*graphic* ↔ **frame** ↔ **relative time**

Performance representation

The VEMUS approach

- a mirror metaphor
- feedback based pedagogy
- score annotation with performance representations

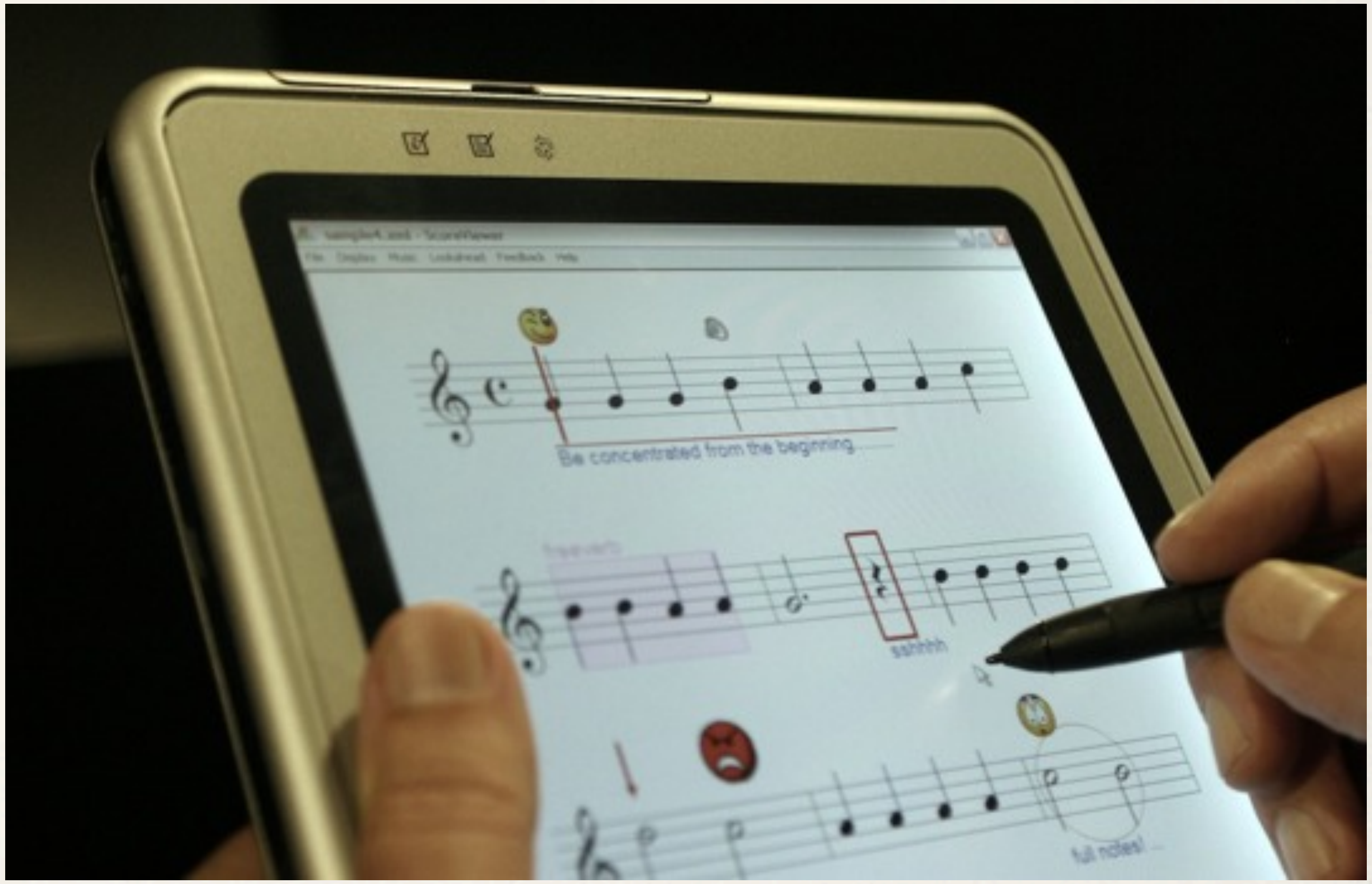


Performance representation

The VEMUS approach

- a mirror metaphor
- feedback based pedagogy
- score annotation with performance representations

- static design,
- tricky to extend,
- awkward to experiment.



Performance representation

Hypothesis

Approach the graphic of a signal as a *graphic signal*.

Performance representation

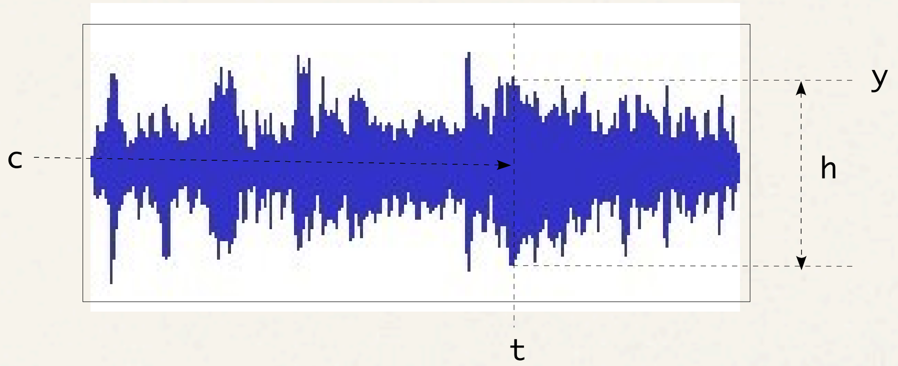
Hypothesis

Approach the graphic of a signal as a *graphic signal*.

A graphic signal

A composite signal made of:

- a y signal
- a thickness signal
- a color signal



Performance representation

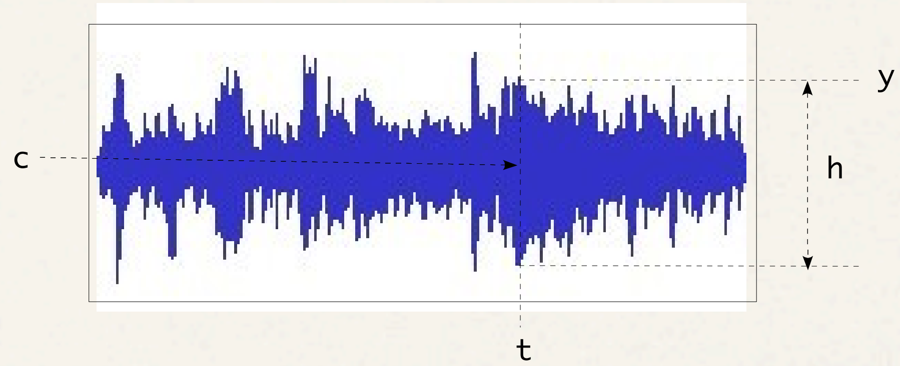
Hypothesis

Approach the graphic of a signal as a *graphic signal*.

A graphic signal

A composite signal made of:

- a y signal
- a thickness signal
- a color signal



Consider a signal S defined as a time function: $f(t) : \mathbb{R} \rightarrow \mathbb{R}^3 = (y, h, c) \mid y, h, c \in \mathbb{R}$

This signal could be directly drawn (i.e. without additional computation)

Performance representation

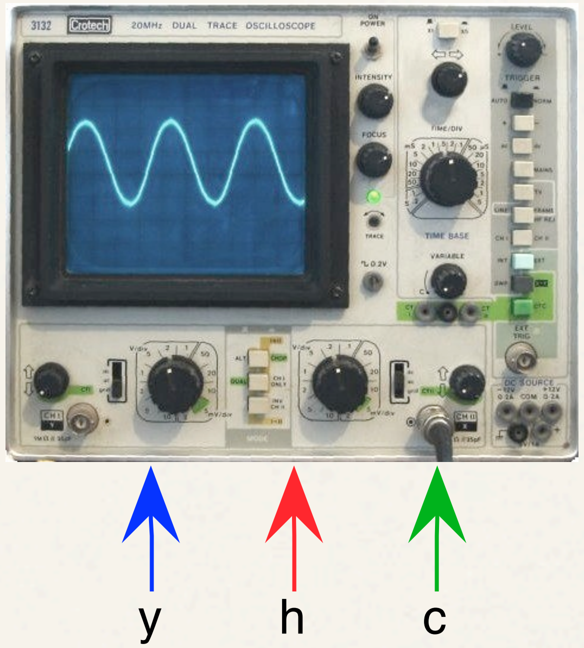
Hypothesis

Approach the graphic of a signal as a *graphic signal*.

A graphic signal

A composite signal made of:

- a y signal
- a thickness signal
- a color signal



Consider a signal S defined as a time function: $f(t) : \mathbb{R} \rightarrow \mathbb{R}^3 = (y, h, c) \mid y, h, c \in \mathbb{R}$

This signal could be directly drawn (i.e. without additional computation)

Performance representation
System expressivity

Examples

$g = S_{f0} \ / \ k_t \ / \ k_c$

S_{f0} : fundamental frequency

k_t : constant thickness signal

k_c : constant color signal

Performance representation
System expressivity

Examples

$g = k_y \ / \ S_{rms} \ / \ k_c$

S_{rms} : RMS signal

k_y : constant y signal

k_c : constant color signal

Performance representation
System expressivity

Examples

$g = S_{f0} \;/\; S_{rms} \;/\; k_c$

S_{rms} : RMS signal

S_{f0} : fundamental frequency

k_c : constant color signal

Performance representation
System expressivity

Examples

$g0 = S_{f0} \;/\; S_{rms0} \;/\; k_c0$

S_{f0} : fundamental frequency

S_{rms0} : f0 RMS values

$g1 = S_{f0} \;/\; S_{rms1} + S_{rms0} \;/\; k_c1$

S_{rms1} : f1 RMS values

$g2 = S_{f0} / \; S_{rms2} + S_{rms1} + S_{rms0} \;/\; k_c2$

S_{rms2} : f2 RMS values

...

$g = g2 \;/\; g1 \;/\; g0$

INScore OSC Messages

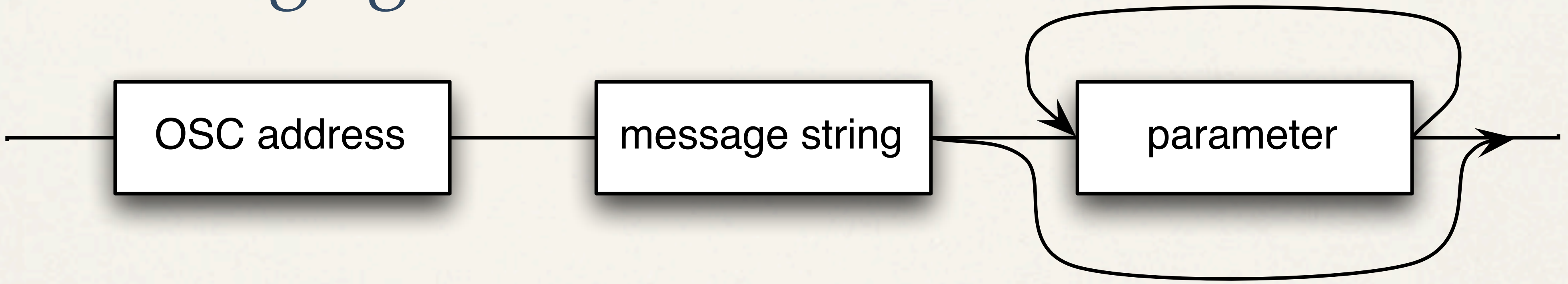
An «object oriented» approach

INScore OSC Messages

An «object oriented» approach

- The OSC address is like an object pointer.
- An OSC message is similar to an object method call.
- The OSC address space is dynamic.

OSC message general format

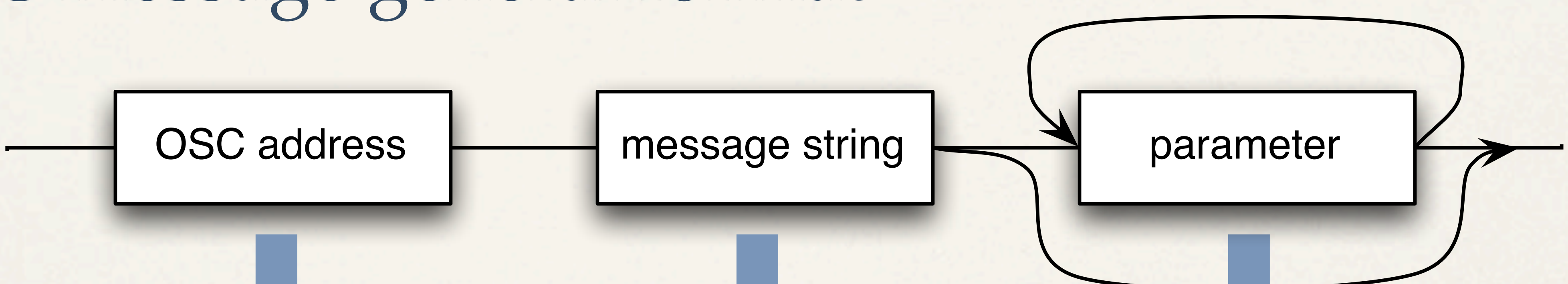


INScore OSC Messages

An «object oriented» approach

- The OSC address is like an object pointer.
- An OSC message is similar to an object method call.
- The OSC address space is dynamic.

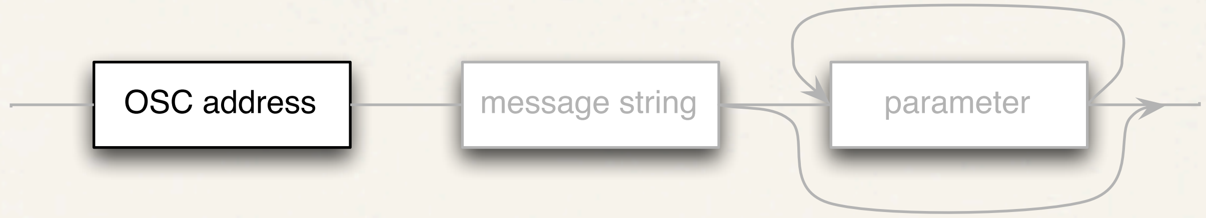
OSC message general format

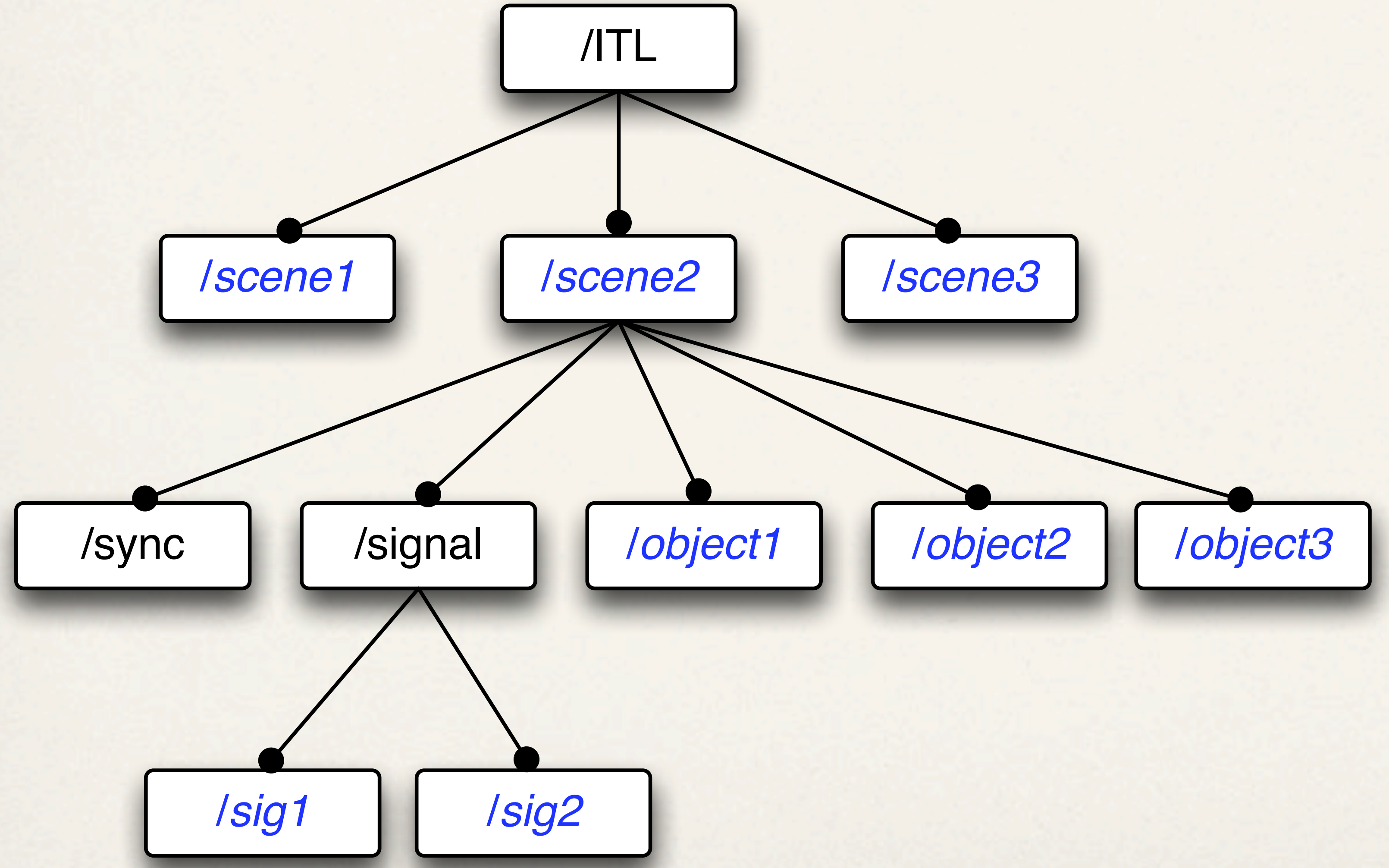


Example

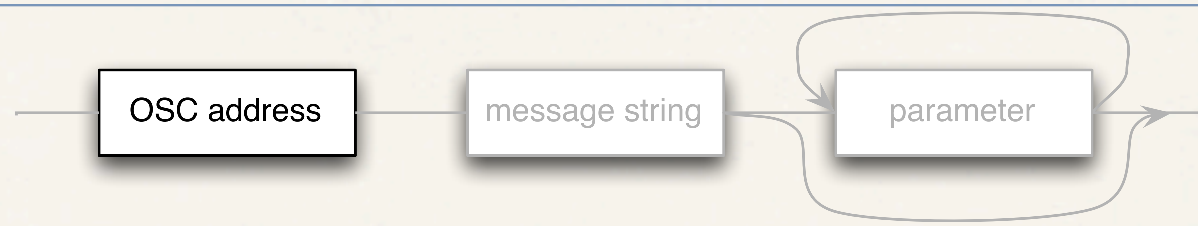
```
/ITL/scene/score   color    255 128 40 150

       score->color(255, 128, 40, 150)
```

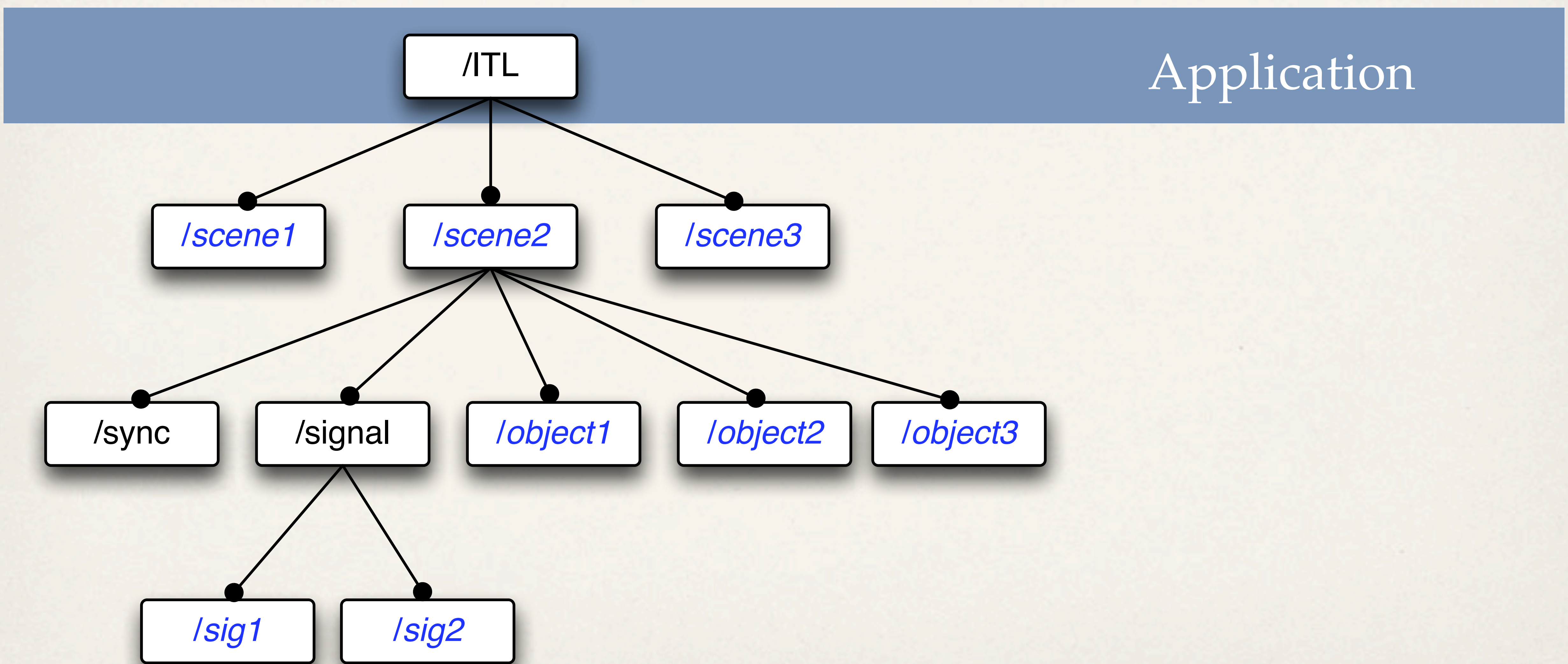
INScore OSC Address Space



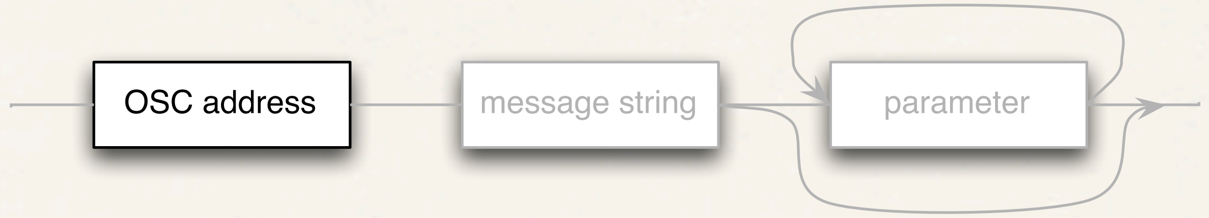


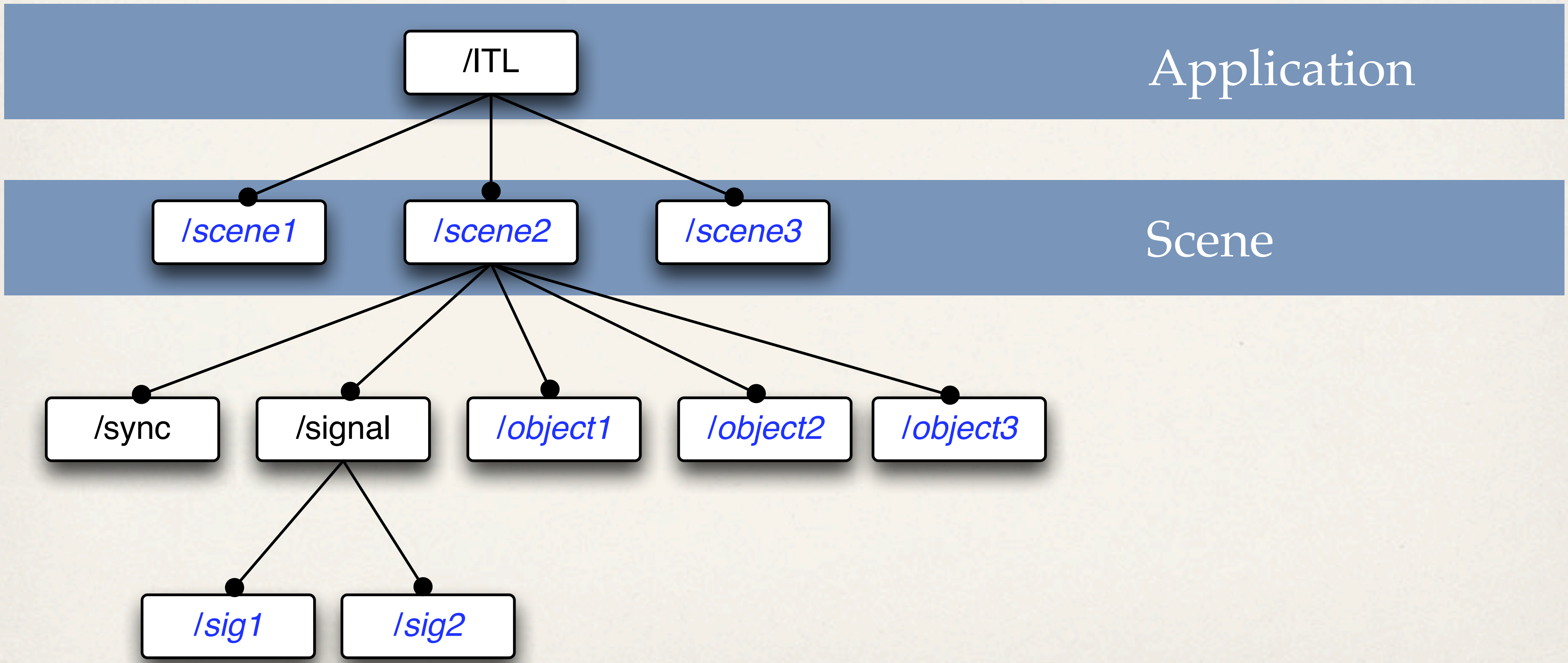
INScore OSC Address Space



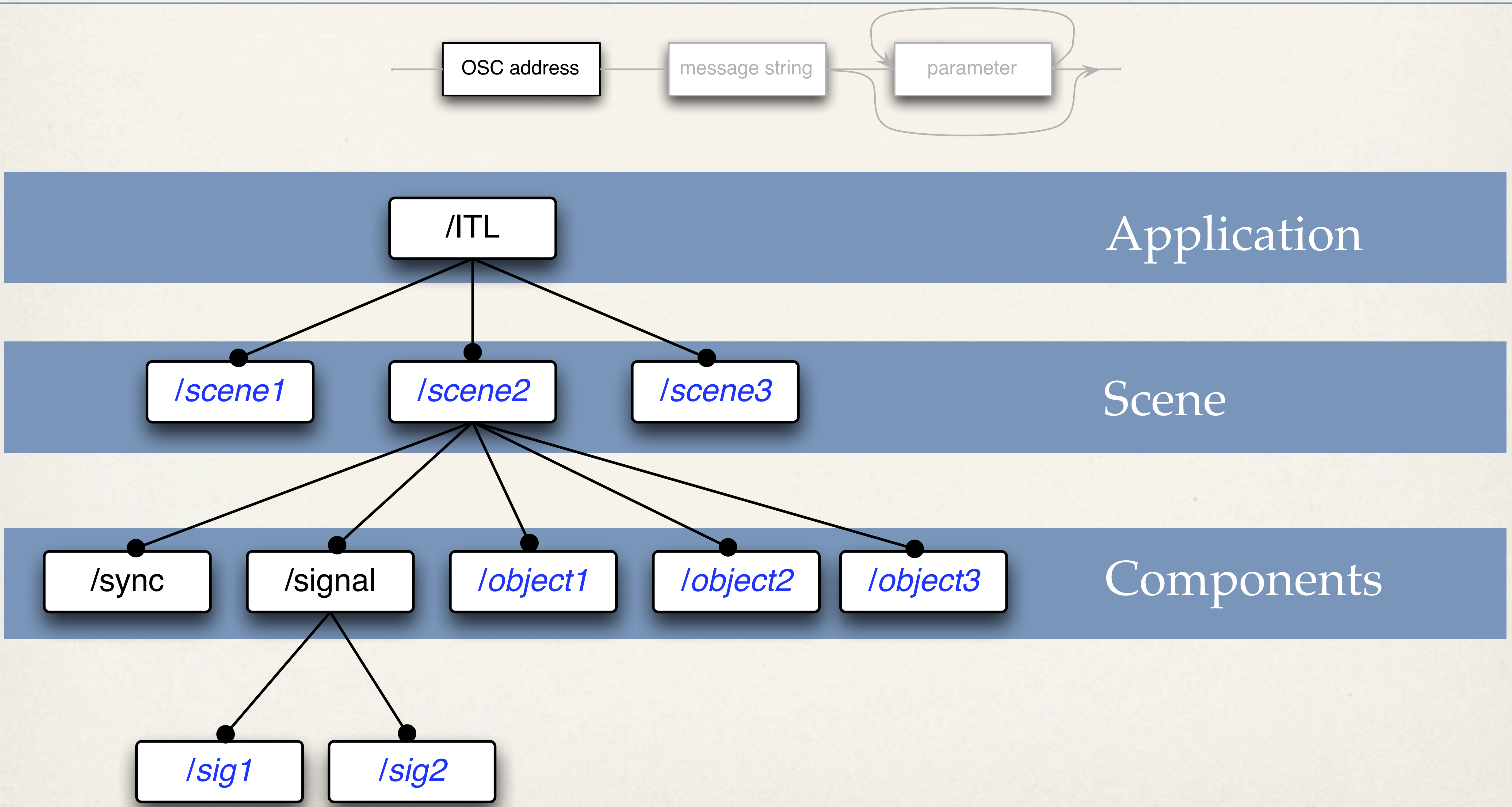


INScore OSC Address Space

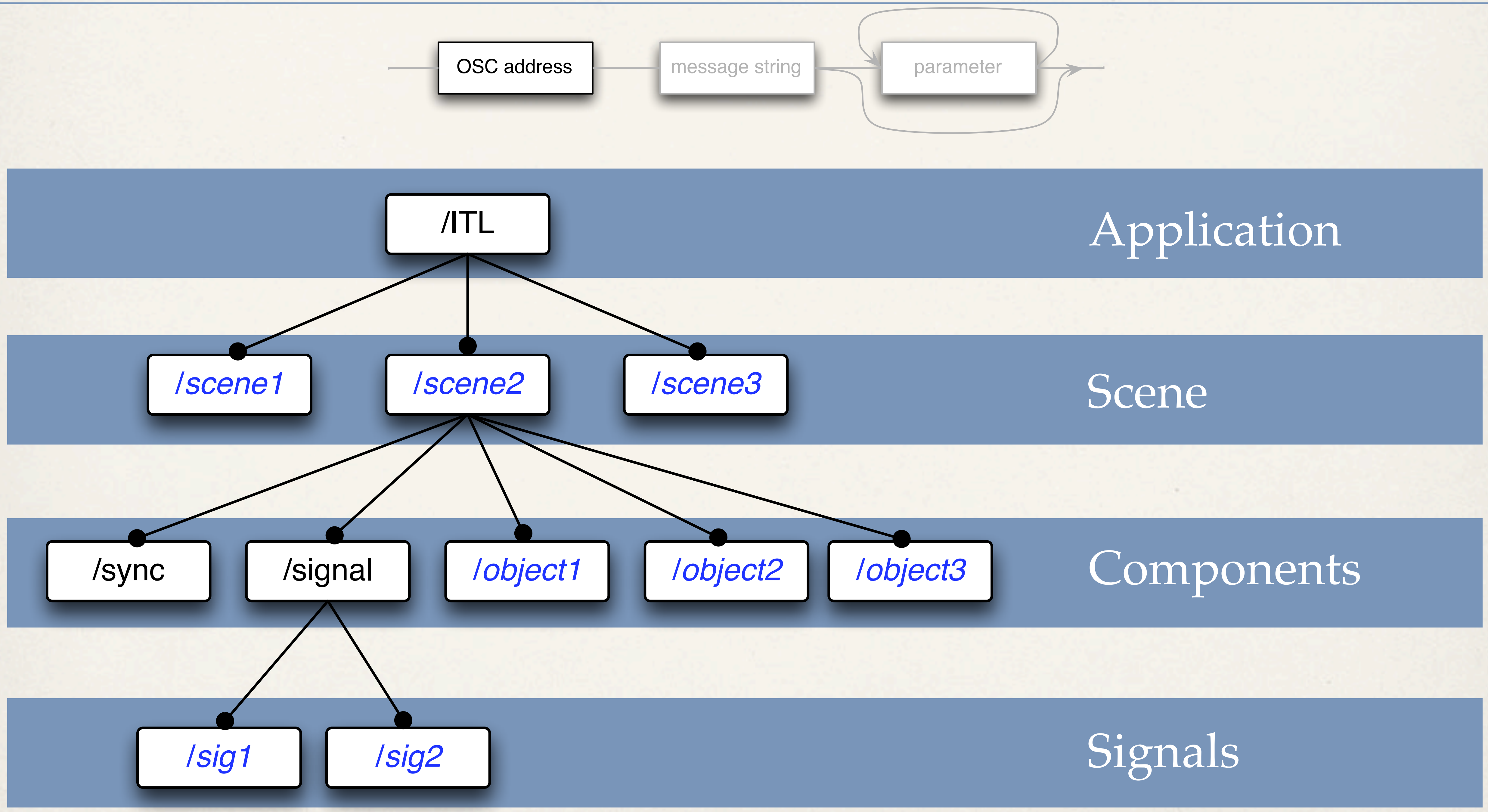




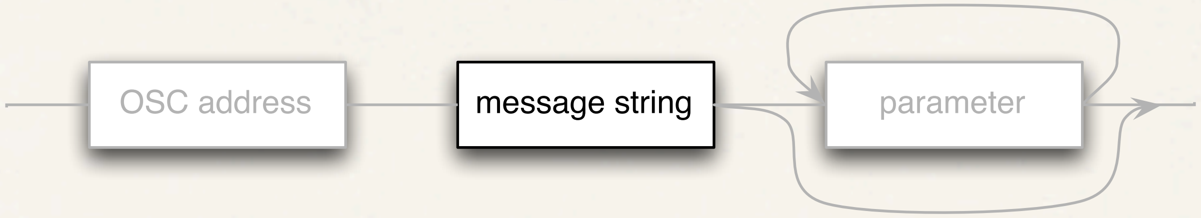
INScore OSC Address Space



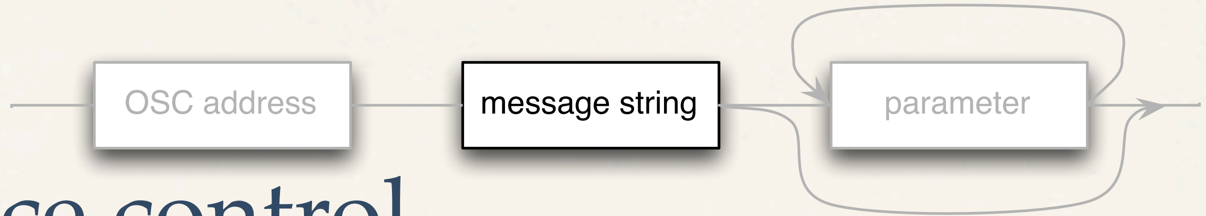
INScore OSC Address Space



Messages Strings



Messages Strings



Graphic space control

- Position:
  `x, y, z, angle, scale, dx, dy, dz, dangle, dscale`
- Color:
  `color, dcolor, red, green, blue, dred, dgreen, dblue, alpha, dalpha, hue, saturation, brightness, dhue, dsaturation, dbrightness`

Time space control

- Time position:
  `date, ddate, clock`
- Duration:
  `duration, dduration`

Constructor

- `set`

Query message

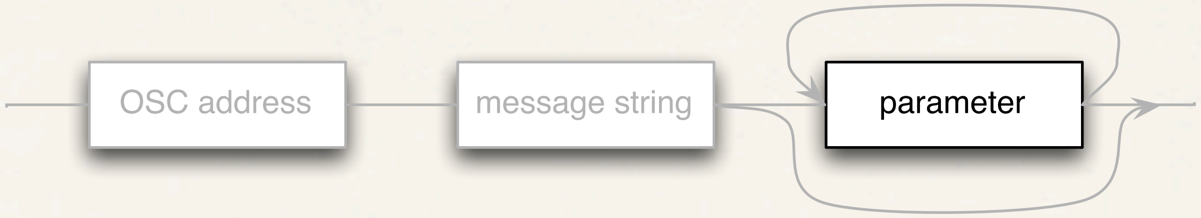
- `get`

Time and graphic spaces relations

- `map`

Signals and graphic signals messages

Messages Parameters



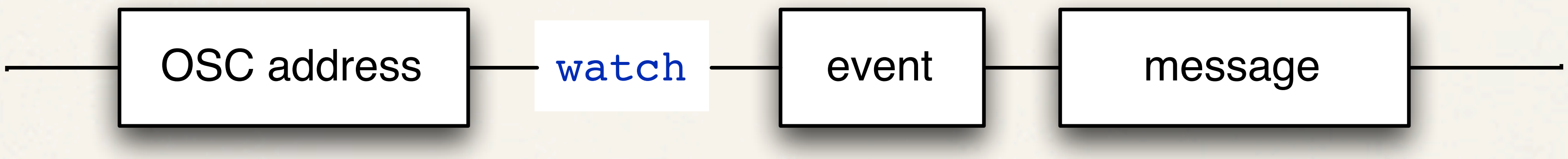
Direct use of basic OSC types

- `int32`
- `float32`
- `OSC-string`

Relaxed types but strict parameters count

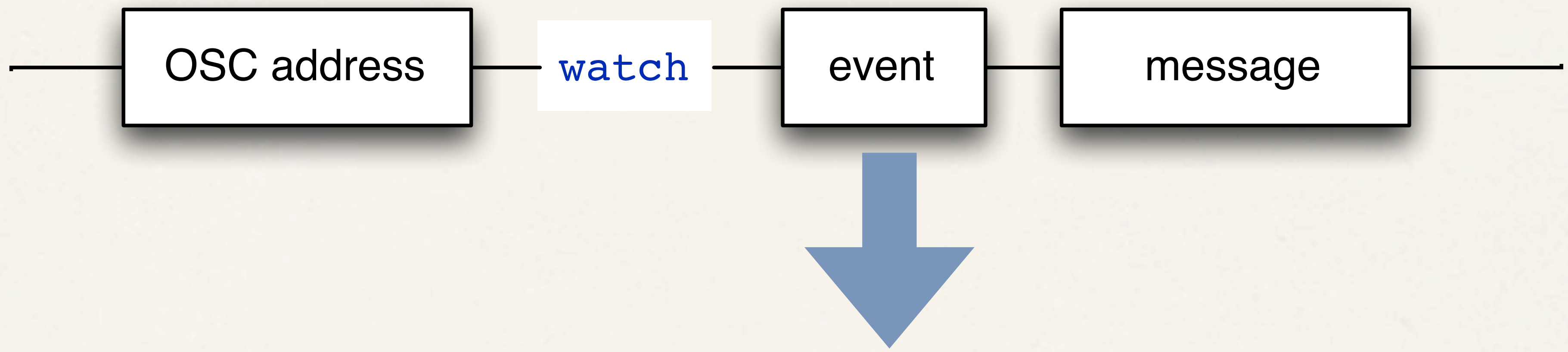
Interaction Messages

Basic principle



Interaction Messages

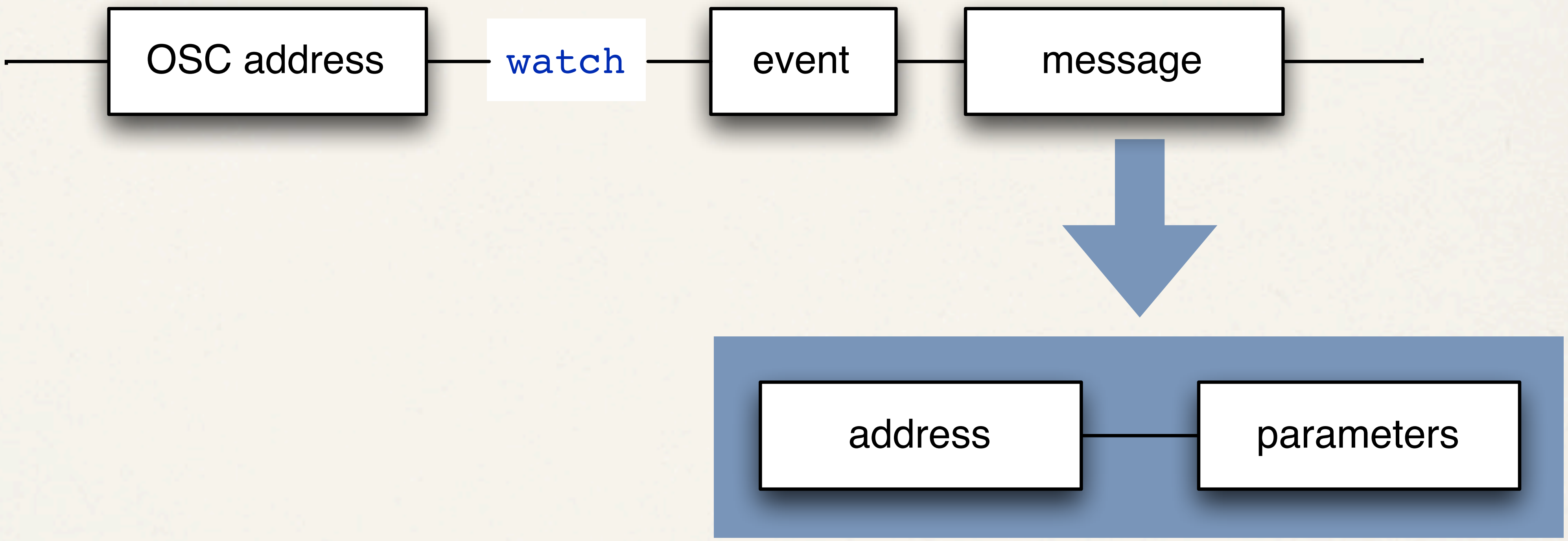
Basic principle



- mouse up, mouse down, mouse move, mouse enter, mouse leave ...
- time enter, time leave

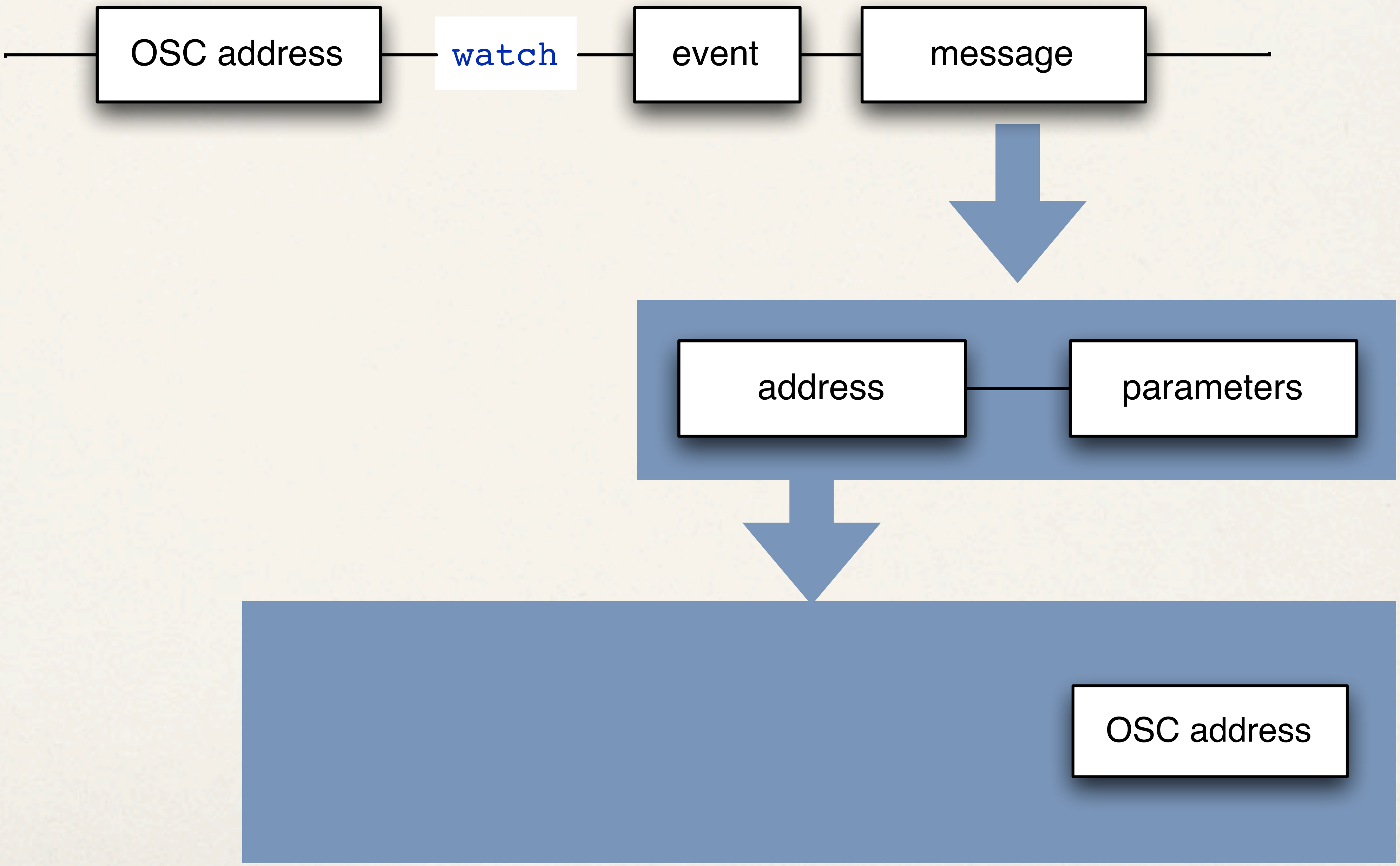
Interaction Messages

Basic principle



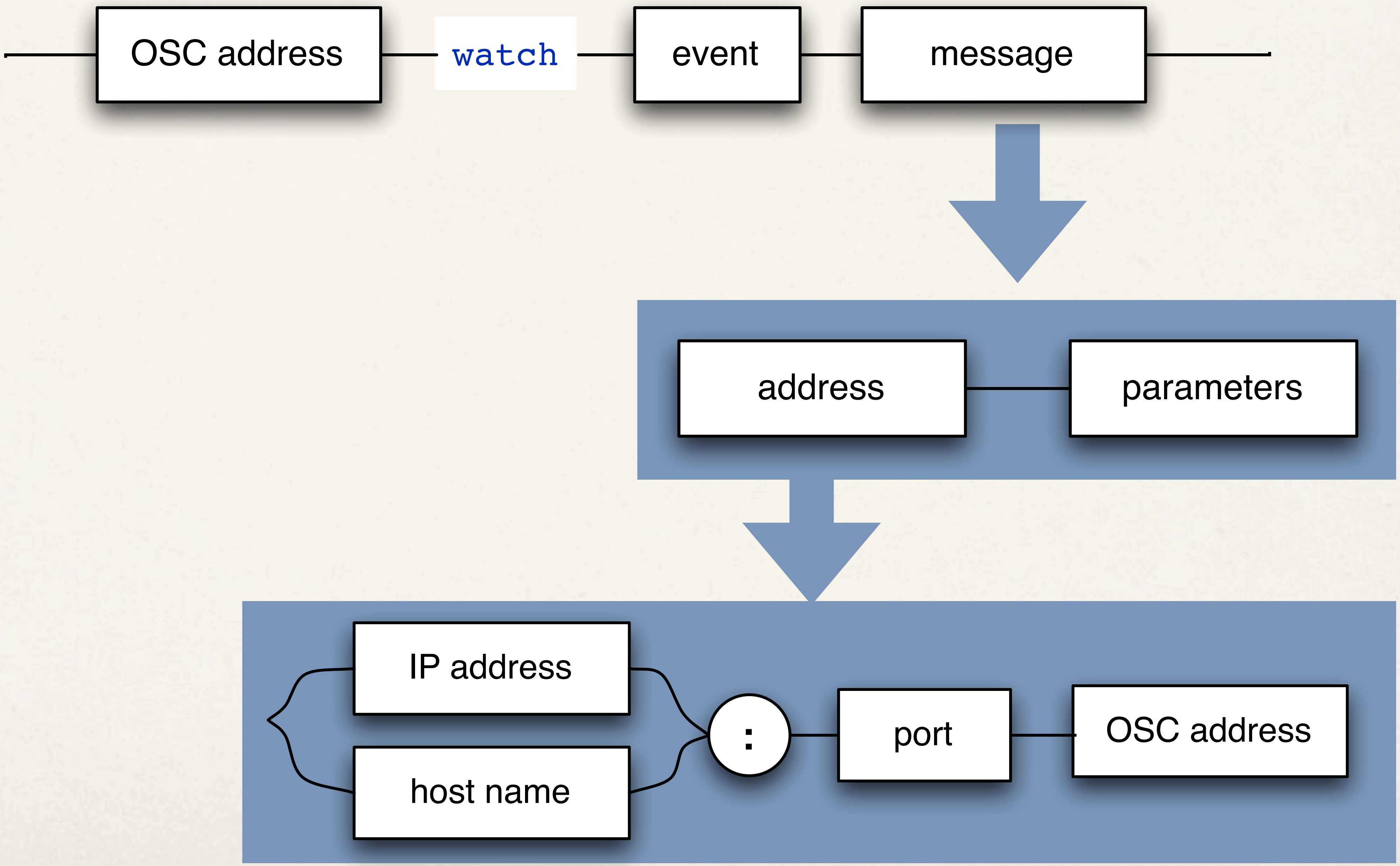
Interaction Messages

Basic principle



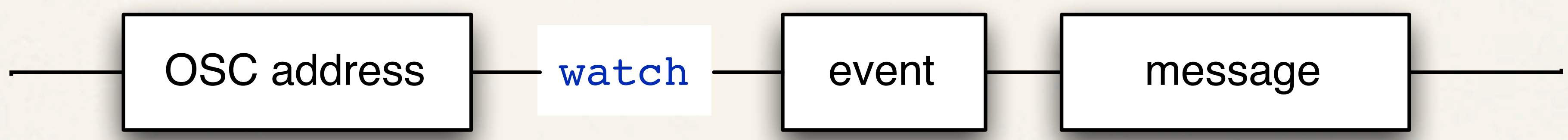
Interaction Messages

Basic principle



Interaction Messages

Basic principle

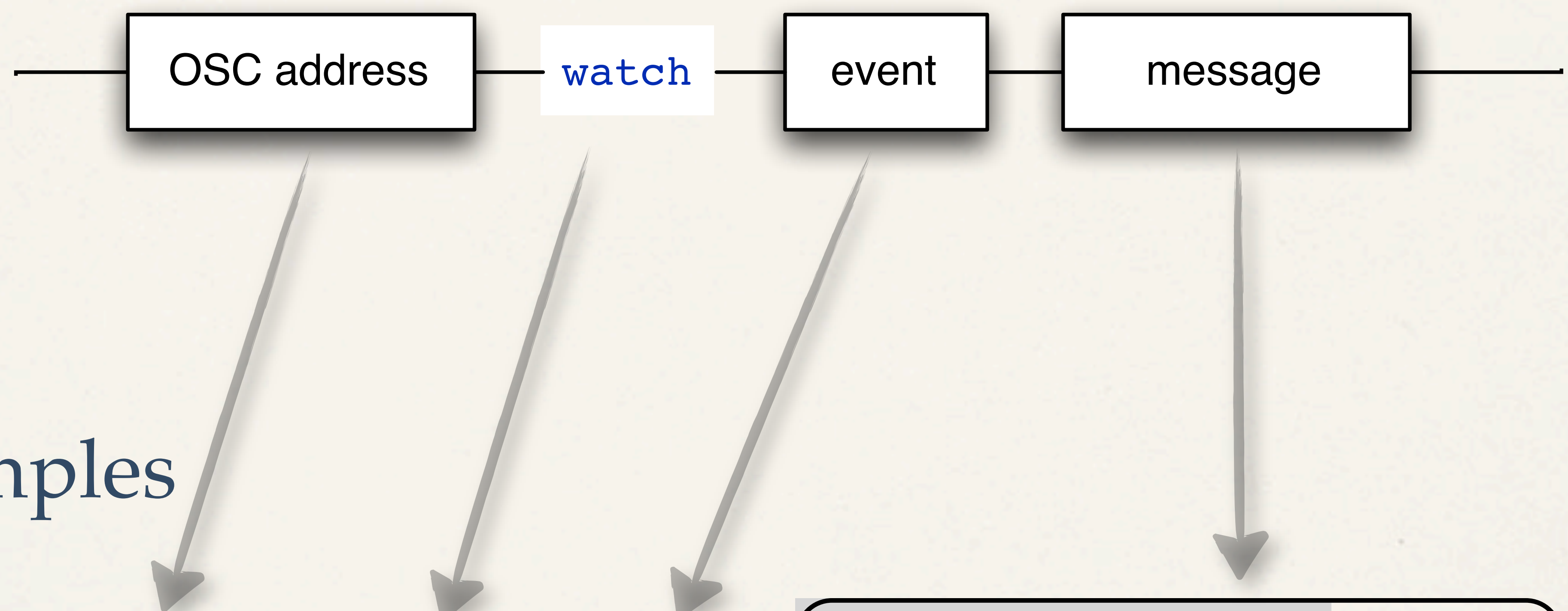


Examples

```
/ITL/scene/myObject watch mouseDown /ITL/scene/myObject show 0
```

Interaction Messages

Basic principle



Examples

```
/ITL/scene/myObject watch mouseDown /ITL/scene/myObject show 0
/ITL/scene/myObject watch mouseDown host.domain.org:12100/an/address start
```

Interaction Messages

Variables

- `$x, $y, $absx, $absy, $sx, $sy`
- `$date`

Address variables

- `$self`
- `$scene`

Message based variables

- `$(a valid INScore ‘get’ message)`

Scaling variable values

- `$x[min, max], $y[min, max]`

Date quantification

- `$date[n/d]`

Interaction Messages

Variables

- `$x, $y, $absx, $absy, $sx, $sy`
- `$date`

Address varia

- `$self`
- `$scene`

Message base

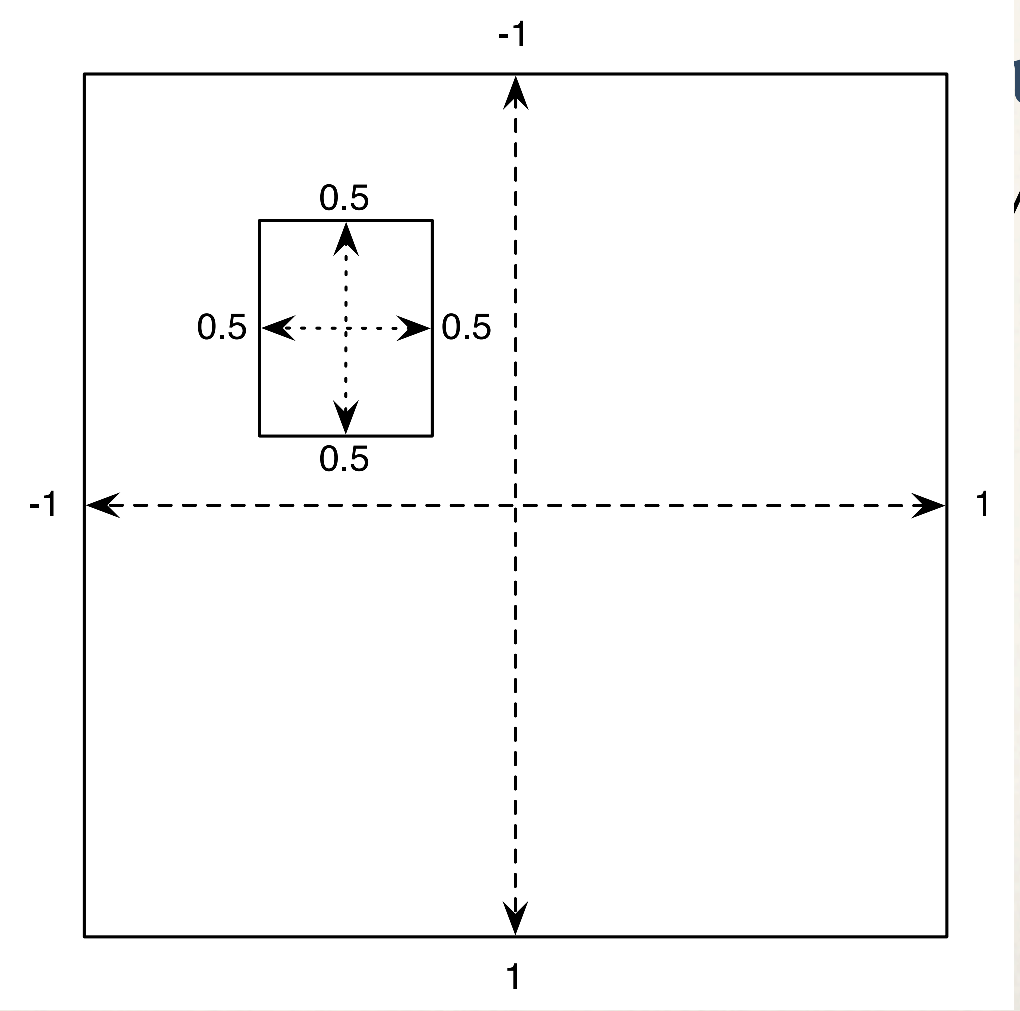
- `$(a valid INScor`

Scaling variable values

- `$x[min, max], $y[min, max]`

uantification

/d]



Interaction Messages

Variables

- `$x, $y, $absx, $absy, $sx, $sy`
- `$date`

Address variables

- `$self`
- `$scene`

Message based variables

- `$(a valid INScore ‘get’ message)`

Scaling variable values

- `$x[min, max], $y[min, max]`

Date quantification

- `$date[n/d]`

Scripting

- INScore files as script files.
- Supports variables
- Javascript support (embedded by default)

```
<?javascript   ... any javascript code ... ?>
```

- optional Lua support (not embedded by default)

```
<?lua   ... any lua code ... ?>
```

http://inscore.sourceforge.net