Networked Comprovisation Strategies with ZScore

Slavko Zagorac
Michael Zbysynski

Goldsmiths, University Of London

Presentation outline

Networked Comprovisation

- decision-making spectrum
- notational perspective
- compositional and technical challenges

Dynamic Notation Strategies in ZScore

- score = data + algorithms
- multiple notation representations (musicians, audience, composer …)
- performance control democratisation
- demo

QnA

Blurred boundaries

Composition

Music score - an incomplete information set

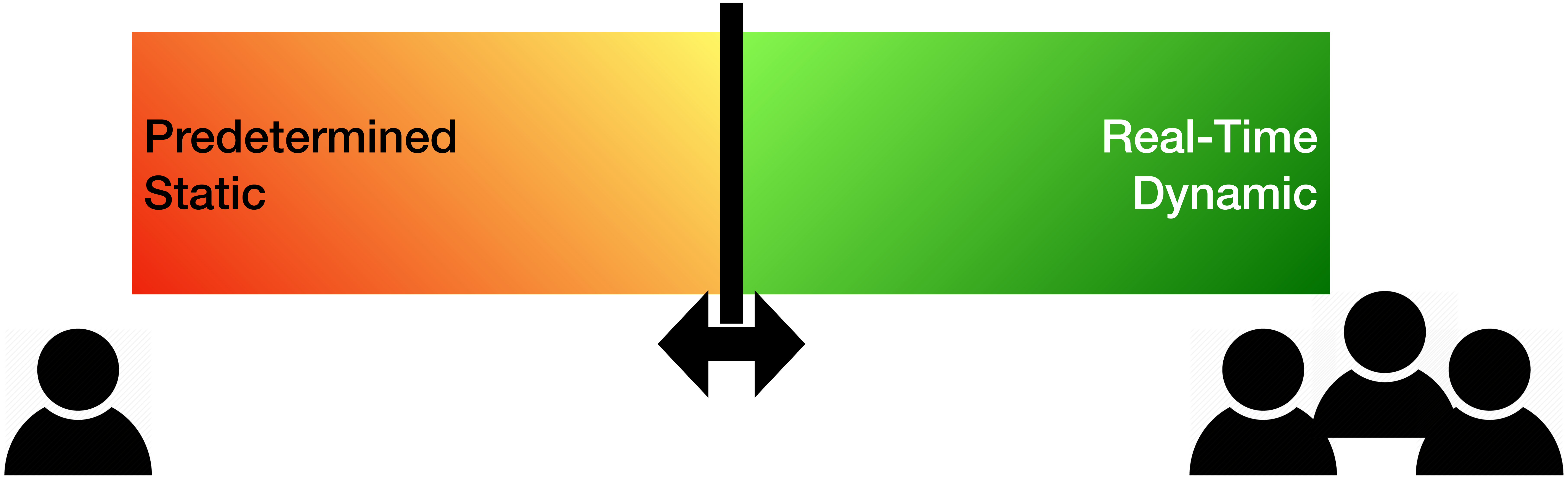
Relies on performer’s understanding of style, aesthetics, playing techniques …

Free Improvisation

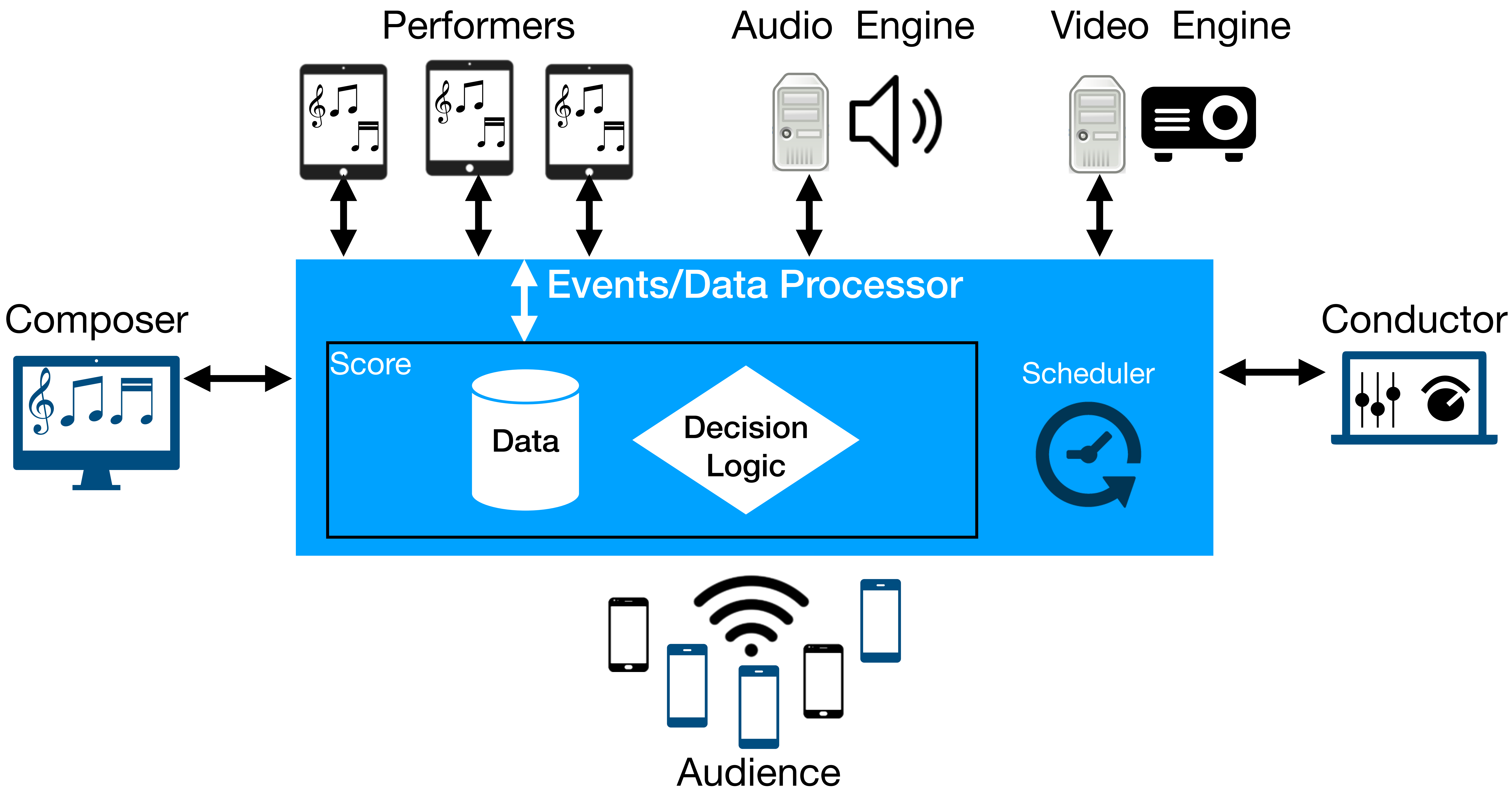
Unwritten degrees of freedom and constraints

Dependent on performer’s experience, learned techniques, performance context …

Comprovisation decision-making spectrum



Event-driven networked performance system



Networked notational perspective

Score as a collection of data and algorithms

- Composition: static data and dynamic behaviour modelling
- Scheduled vs triggered events
- Event timing constraints (update window)
- Contingent nature of dynamic notation

Multiple score representations

- Different participant types, heterogeneous clients
- Any participant can be assigned decision-making agency
- Blurred traditional boundaries between
  - composer/conductor/performer/audience
- Akin to ancient communal music-making (musicking)

Networked performance challenges

Technical

- Scheduling and synchronisation over a network
- Performance: client/server load, latency (wifi vs ethernet)
- Simplicity vs efficiency: stateless clients vs shared state

Practical

- Composition complexity
- Interactive notation layout usability, musicians’ preparation time
- Participant action acknowledgment

Aesthetic

- Does the outcome match participants’ intentions?

Dynamic notation strategies in ZScore

ZScore:

- Networked notation system for mixed ensemble composition and performance
- Mixture of third-party and proprietary software
    - Music Notation: Adobe Illustrator + javascript plugins
    - Scheduling and distribution engine: proprietary (Java)
    - Performance controls GUI: proprietary (Java)
    - Musicians’ front end: INScore
    - Audience front end: any browser (html, javascript, gsap)
    - Audio engine: any OSC (MAX, Supercollider…)
    - Video engine: any OSC (Jitter, INScore…)

Music notation layout in ZScore

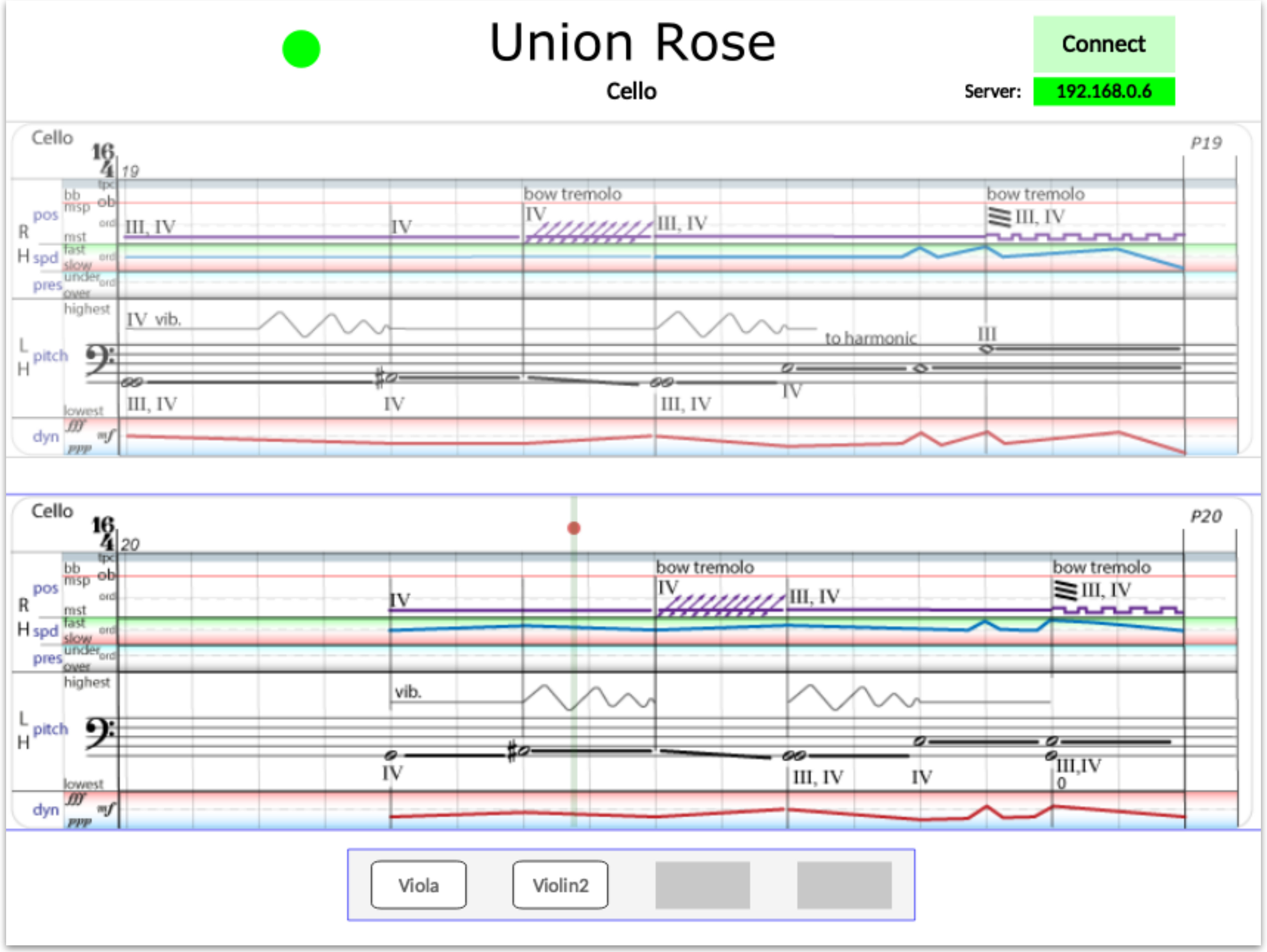
Alternating pane notation strategy

Two panes:

    active and preparatory

Notation not moving,
read traditionally:
left-to-right
top-to-bottom

Update time window
Supports any notation type
Actions displayed below notation

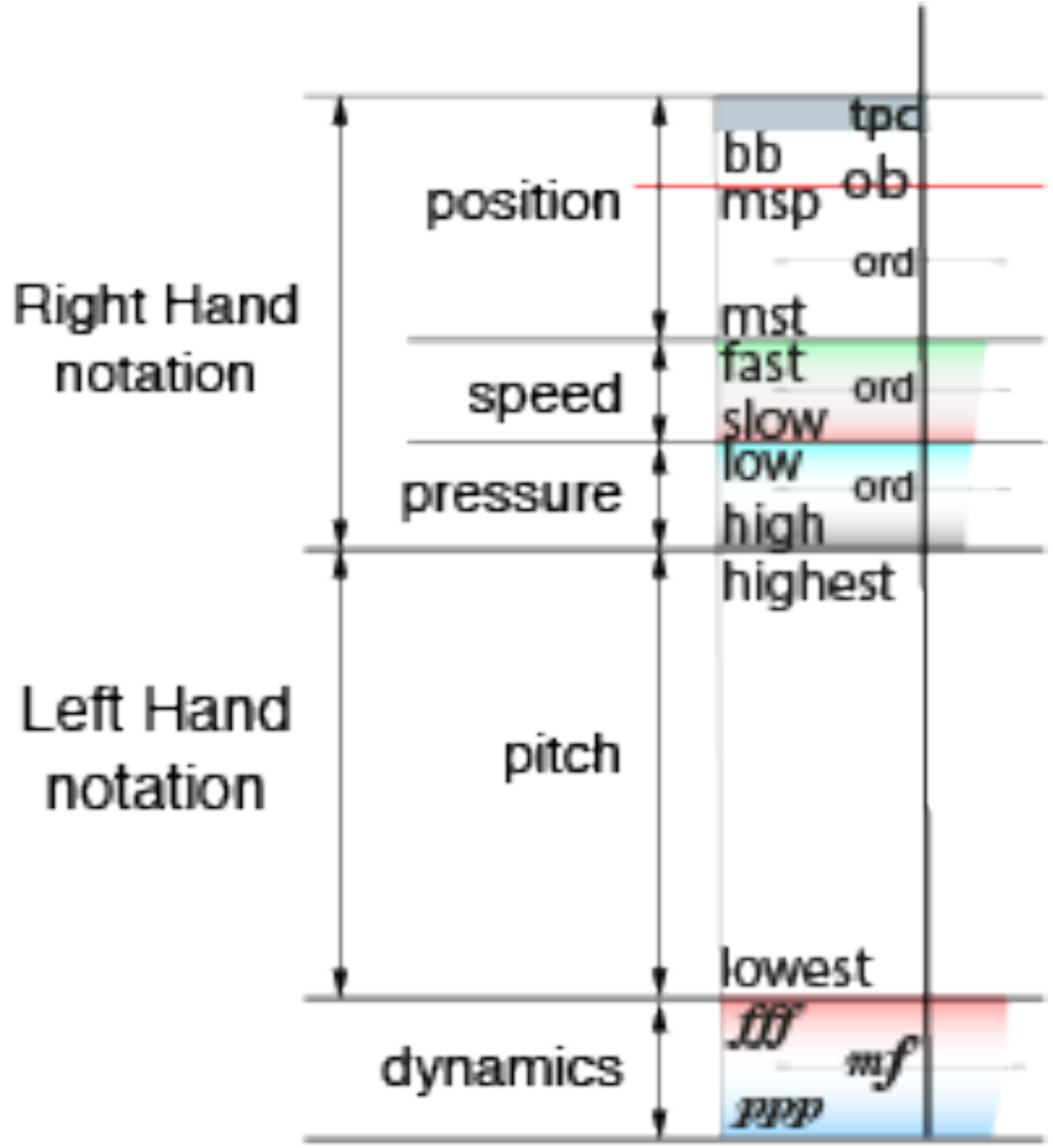


Dynamic performance parameter notation

Vertical separation of playing technique parameters

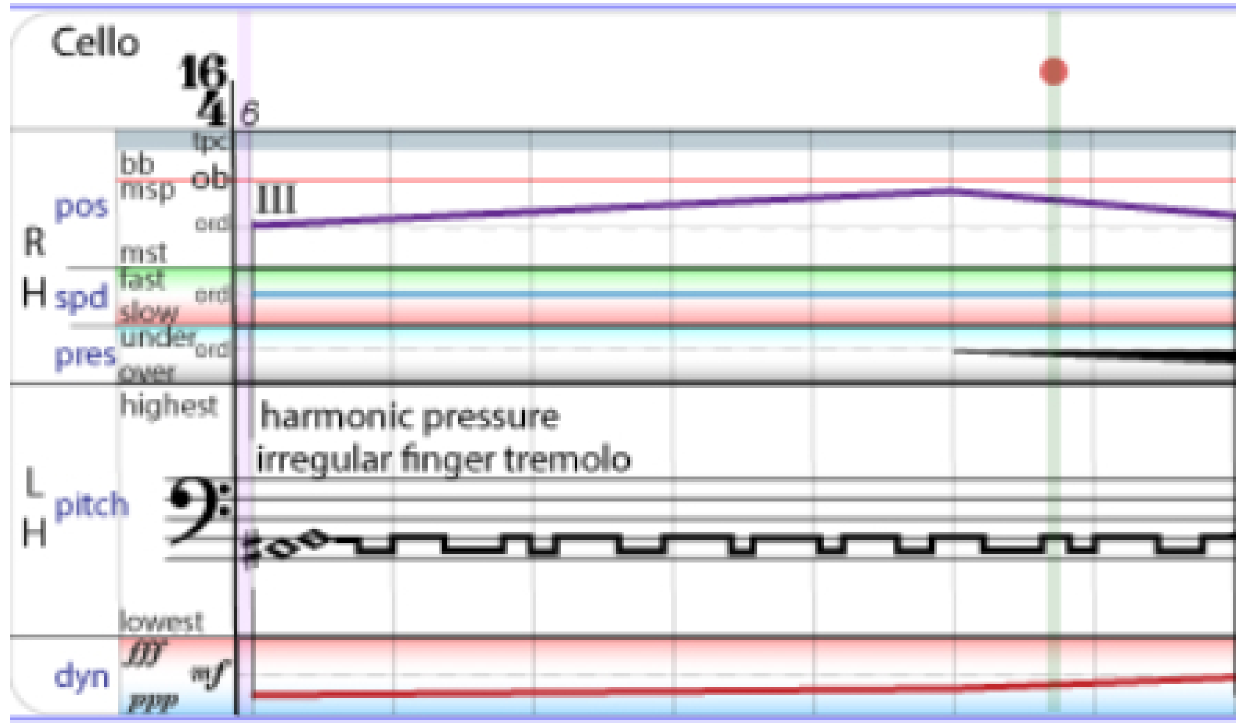
Each parameter that receives dynamic notation has dedicated two-dimensional space where
X axis: time (common for all params)
Y axis: parameter value range

User trials determined optimal placements: e.g. dynamics always below the pitch notation…

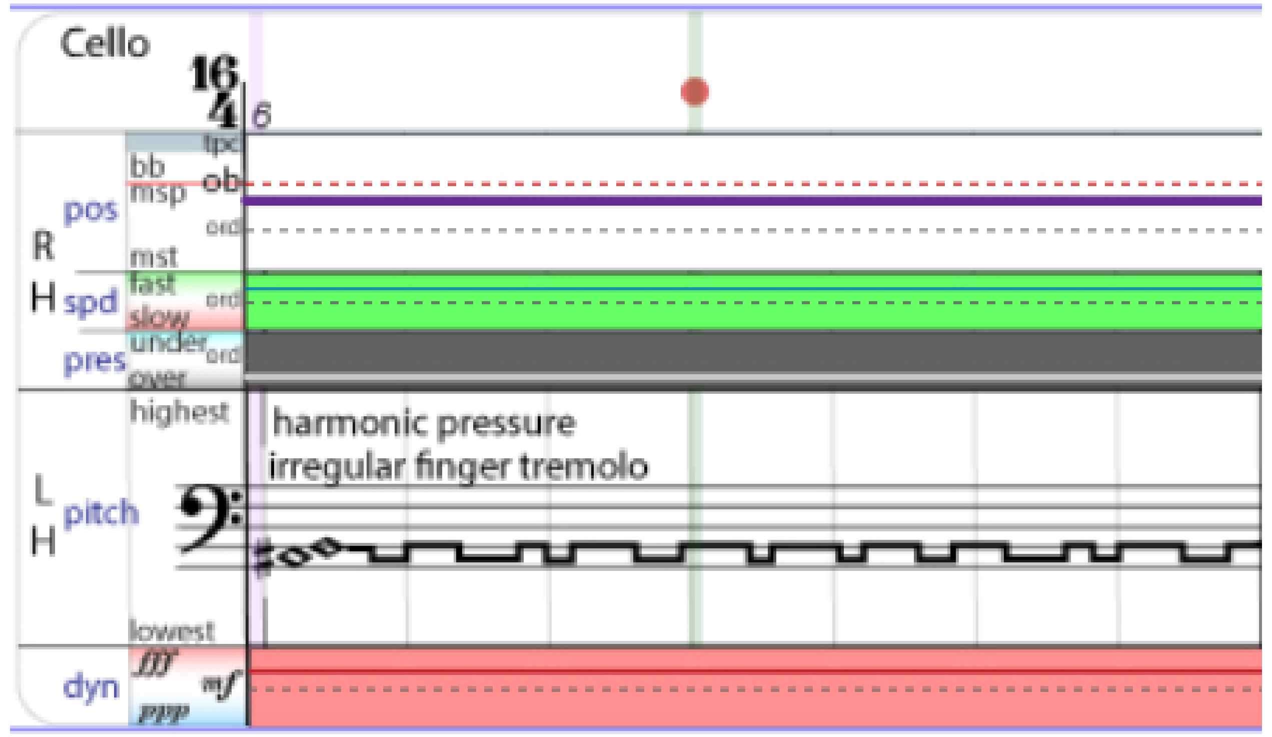


Dynamic notation overlays

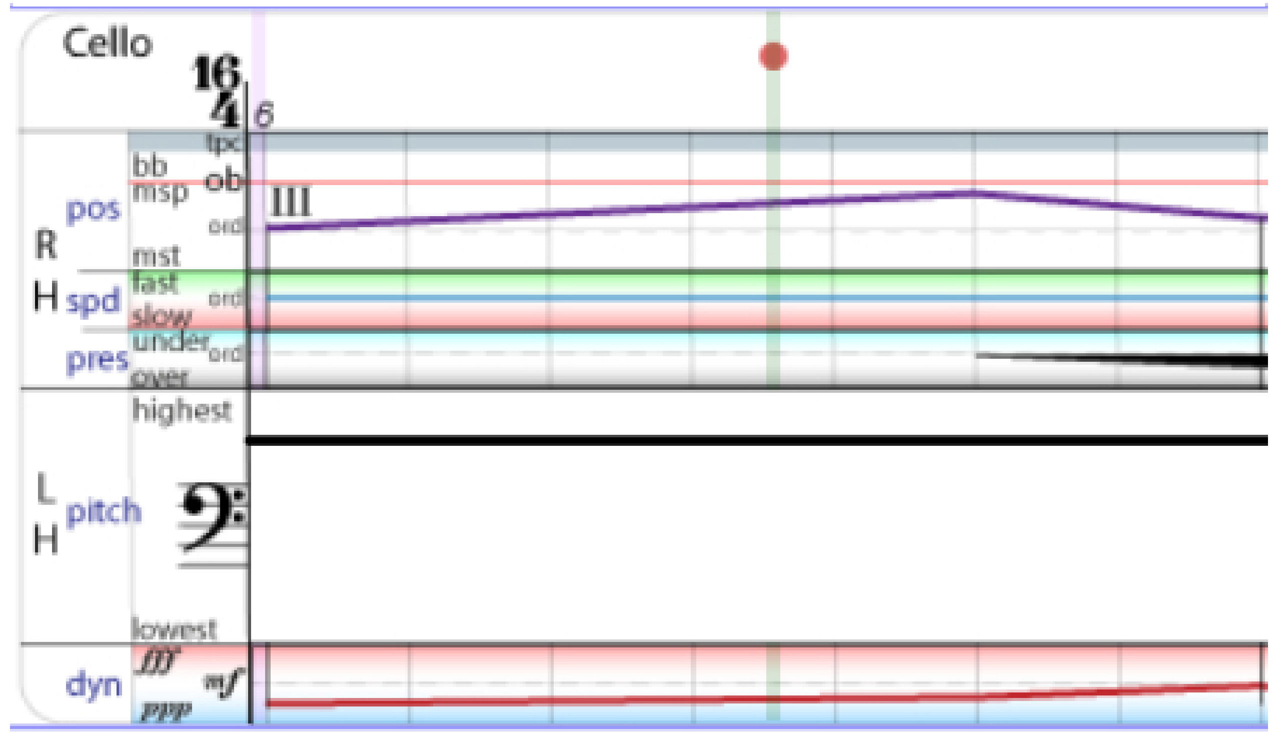
Original notation



Dynamic overlays on for position, speed, pressure and dynamics



Dynamic overlay on for pitch only

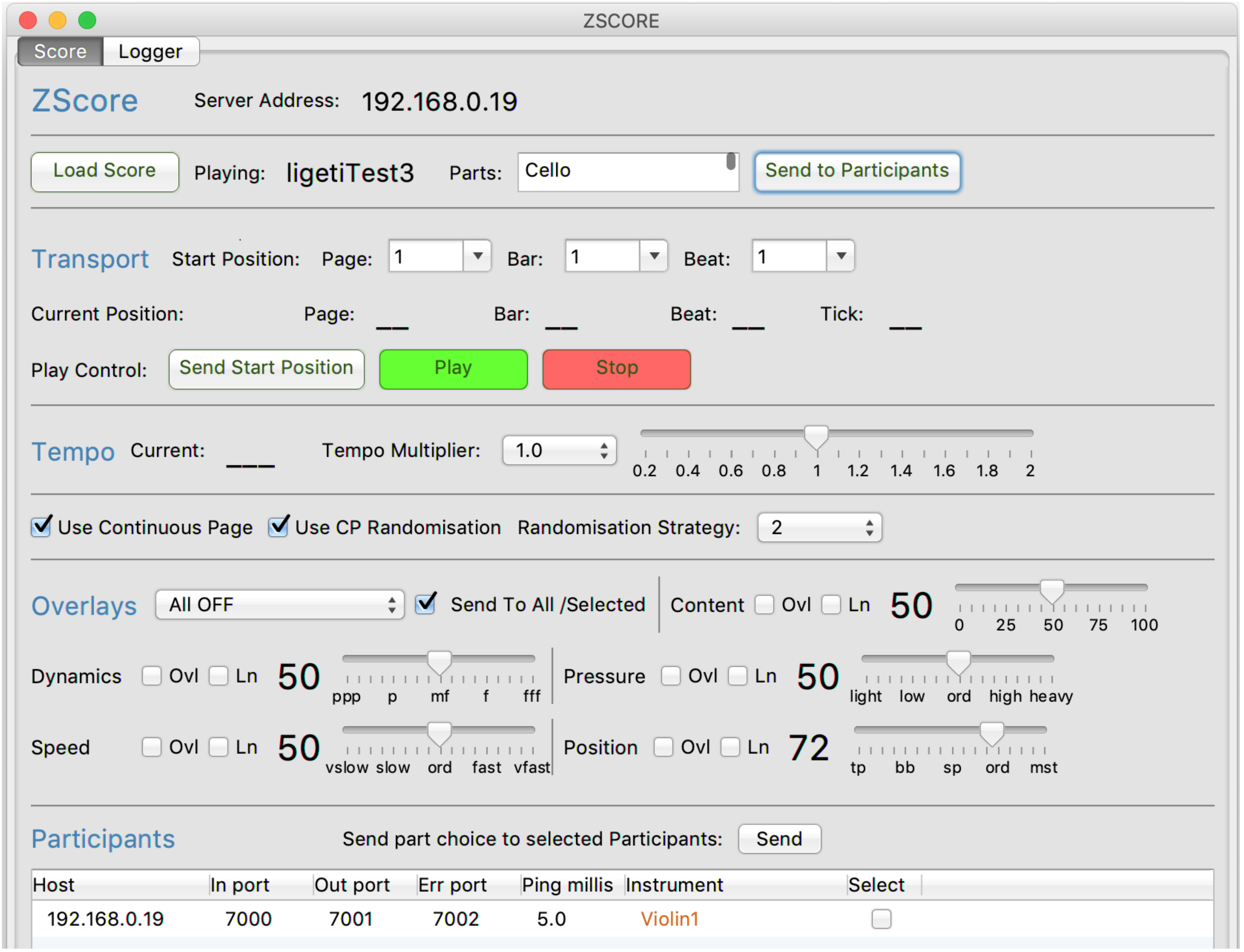


Dynamic notation controls in ZScore

Java client

Currently allows for control of:

- Current position
- Tempo
- Dynamic overlays
- Randomisation strategy
- Participant selection



ZScore Server Address: 192.168.0.19

Playing: ligetiTest3 Parts: Cello

Transport Start Position: Page: 1 Bar: 1 Beat: 1

Current Position: Page: __ Bar: __ Beat: __ Tick: __

Play Control:

Tempo Current: ___ Tempo Multiplier: 1.0

☑ Use Continuous Page ☑ Use CP Randomisation Randomisation Strategy: 2

Overlays All OFF ☑ Send To All /Selected

Content □ Ovl □ Ln 50

Dynamics □ Ovl □ Ln 50

Pressure □ Ovl □ Ln 50

Speed □ Ovl □ Ln 50

Position □ Ovl □ Ln 72

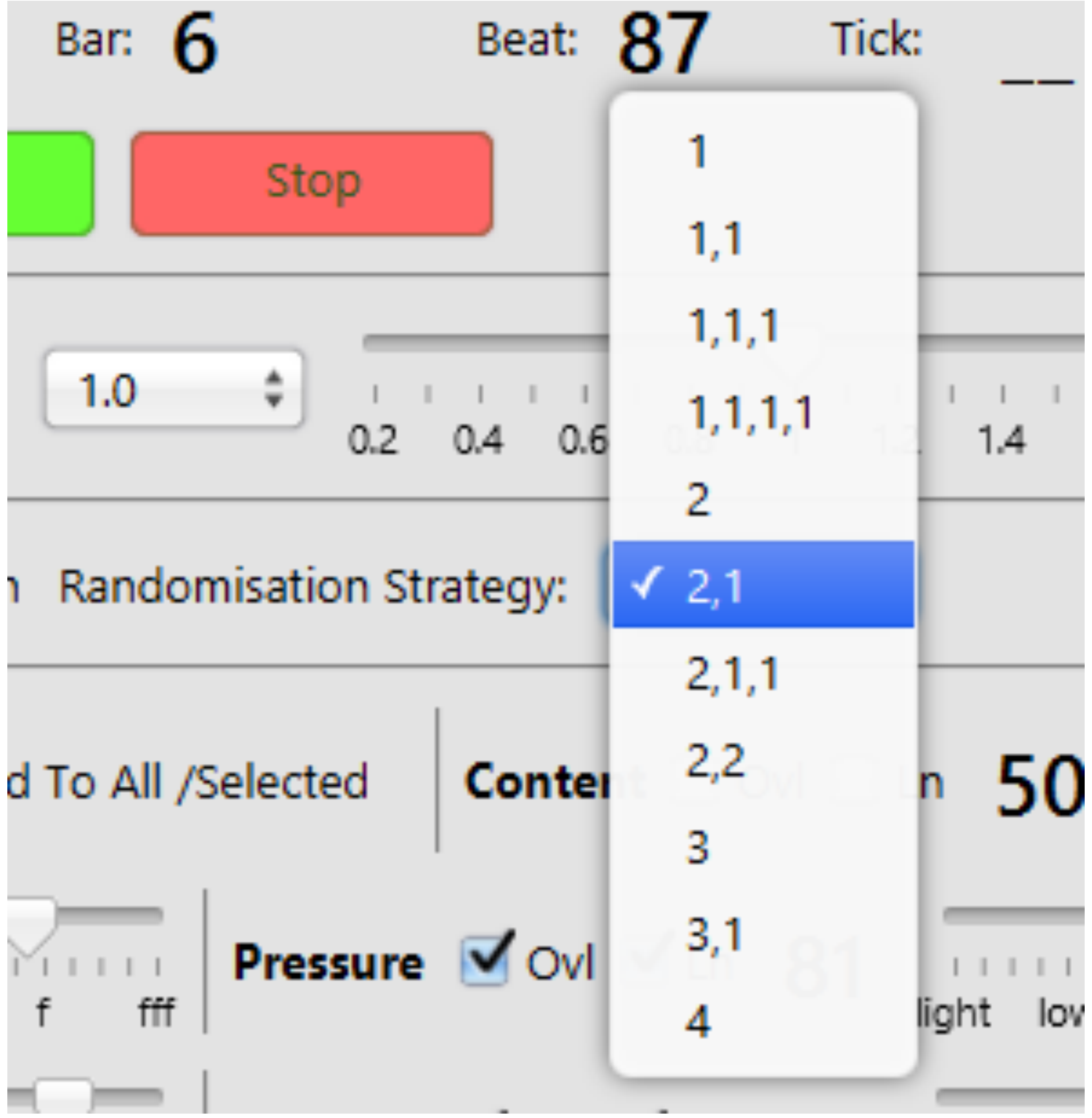
Participants Send part choice to selected Participants:

Host	In port	Out port	Err port	Ping millis	Instrument	Select
192.168.0.19	7000	7001	7002	5.0	Violin1	□

Score randomisation strategies

Randomisation strategies are currently used to select what notation and instrumentation should be used in the next time window

Configuration examples:
‘2’ - two randomly selected instruments play the same randomly selected page
‘2,1’ - two instruments play the same page, one instrument plays another random page
‘1,1,1,1’ - four instruments play different randomly selected page



Bar: 6 Beat: 87 Tick:

Randomisation Strategy: ✓ 2,1

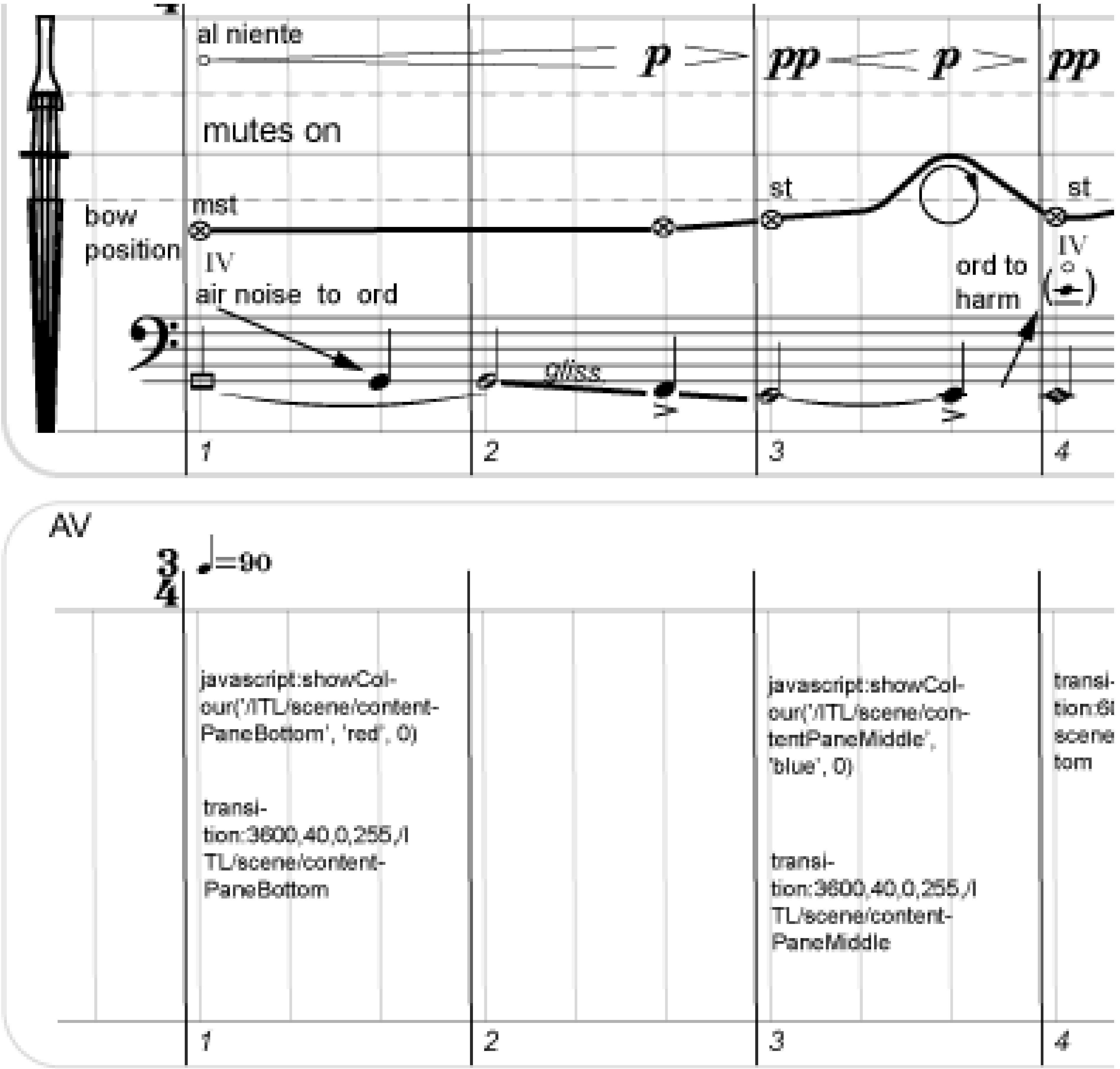
Embedded scripting in ZScore

Scripts can be embedded directly in the score

Each script has associated execution time (beat) and destination

Currently available types:

- Server side javascript
- Client side javascript (web)
- OSC (INScore, MAX)



```
javascript:showColour('/ITL/scene/contentPaneBottom', 'red', 0)

transition:3600,40,0,255,/ITL/scene/contentPaneBottom
```

```
javascript:showColour('/ITL/scene/contentPaneMiddle', 'blue', 0)

transition:3600,40,0,255,/ITL/scene/contentPaneMiddle
```

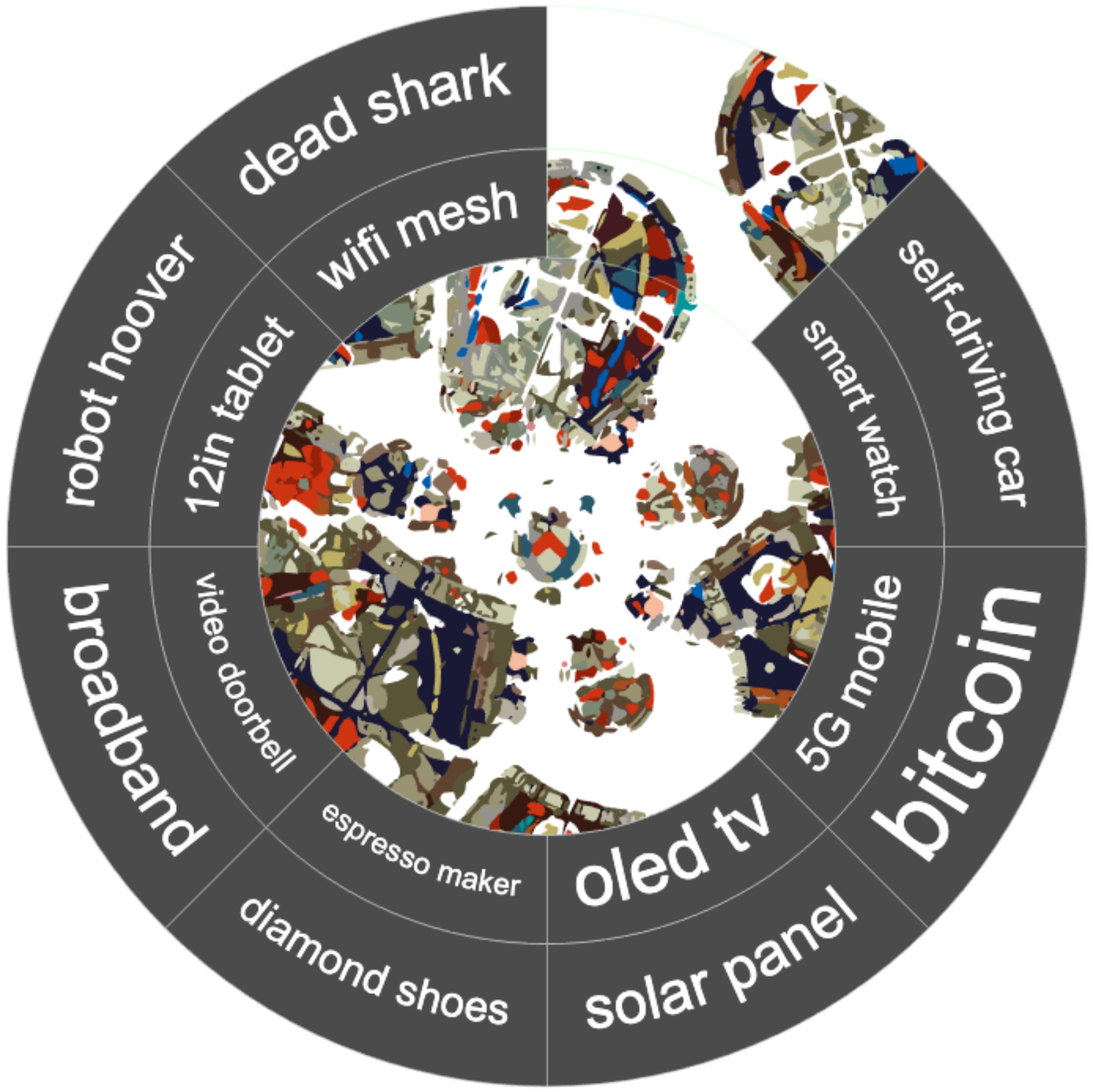
```
transi-
tion:6
scene
tom
```

Audience score representation (web view)

Audience can connect via any internet browser. Implemented in html and javascript.

The score representation can contain static and animated graphics (gap), textual info, audio content and actionable controls.

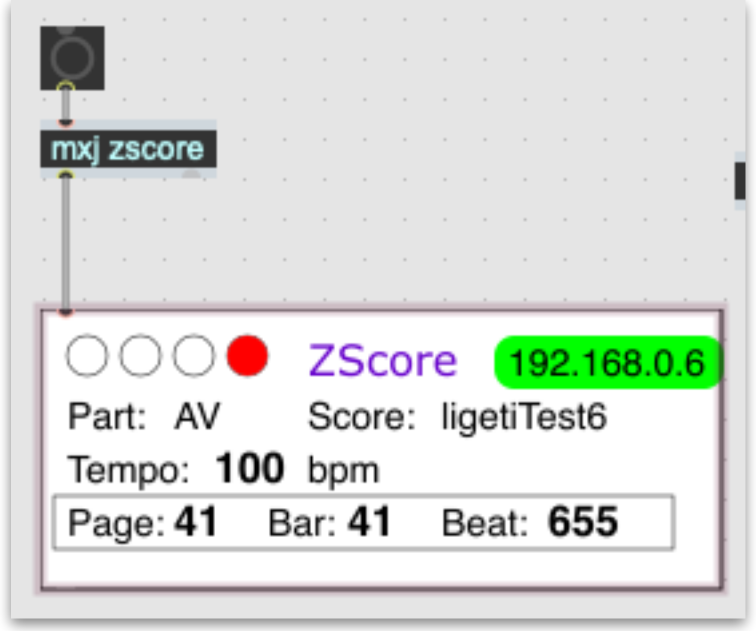
Supports WebSockets, SSE (ever side events) and HTTP polling.



ZScore Max interface

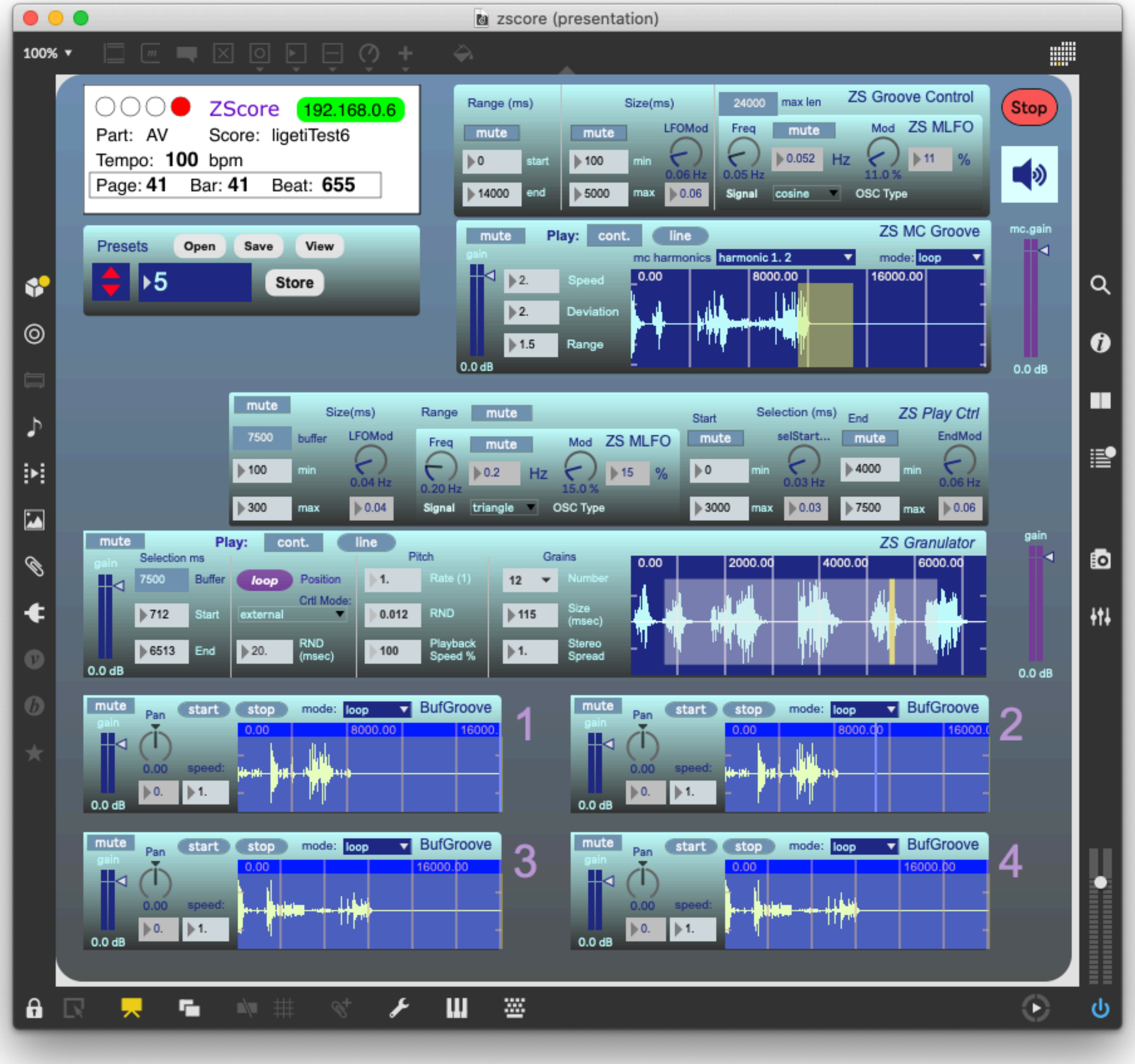
Max interface consists of Java external and JSUI component.

It receives OSC messages defined in the score and distributes to named max objects.



mxj zscore

○○○● ZScore 192.168.0.6
Part: AV Score: ligetiTest6
Tempo: 100 bpm
Page: 41 Bar: 41 Beat: 655



zscore (presentation)

○○○● ZScore 192.168.0.6
Part: AV Score: ligetiTest6
Tempo: 100 bpm
Page: 41 Bar: 41 Beat: 655

Demo

Future Work

Real-time svg notation

Various third party systems integration

- Conductor gesture (Wikinator)
- Drawsocket

Larger audience load/scalability testing

Perpetual usability enahancements

Questions?

email slavko@zagorac.com

web http://www.zagorac.com