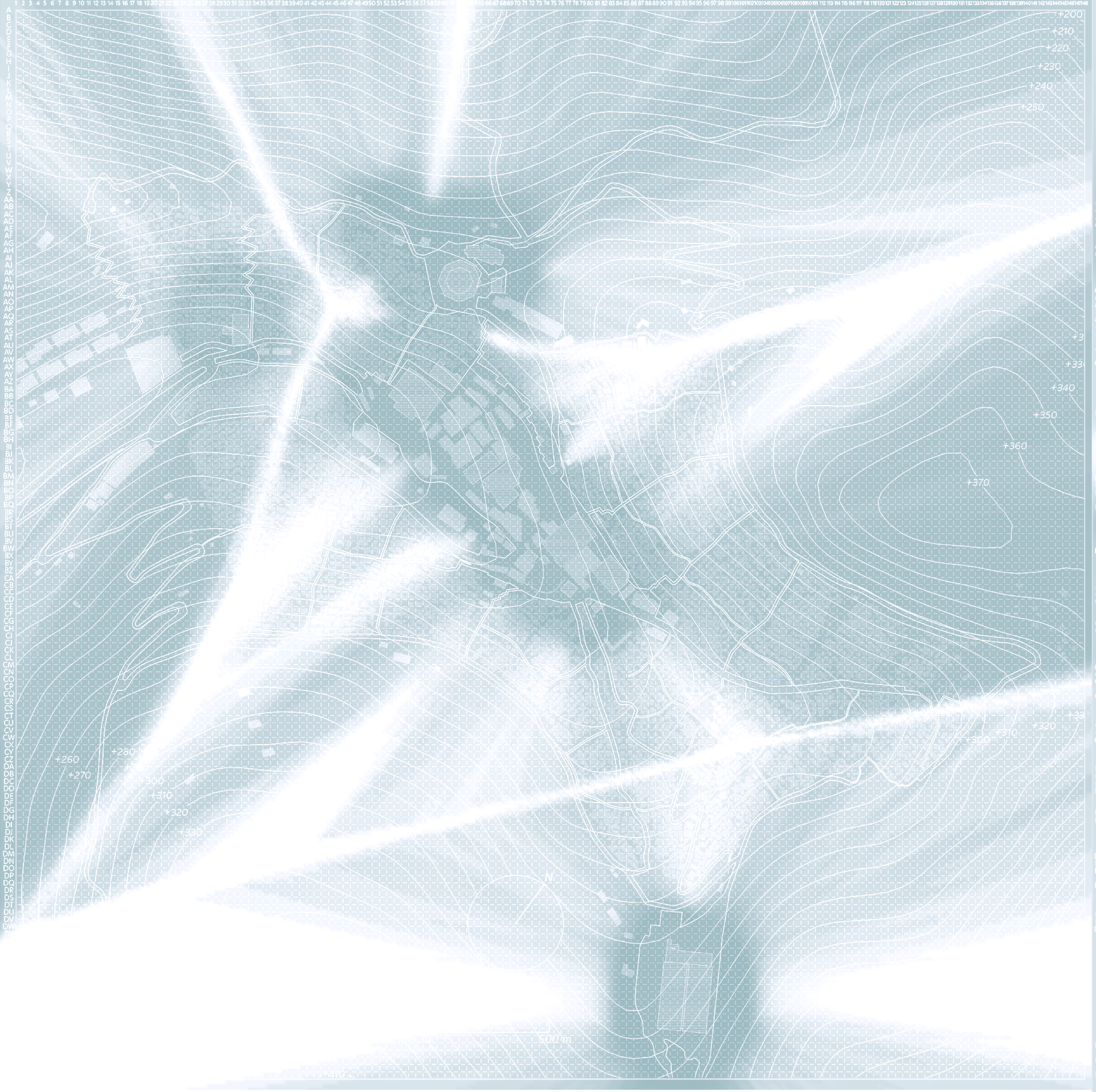


Zhuyuan Liu
Portfolio

MArch / RIBA Part 2
AA School of Architecture
zhuyuan.liu@outlook.com

Table of Contents

1. Sacred Intelligence

Spatial Systems of Cultural Practice

1.1 Site Analysis & Spatial Simulation

Context
Site Analysis
OODA-Based Stakeholder Modelling
Spatial Simulation

Architectural decisions at Larung Gar emerge from the interaction of physical conditions, self-organised development and competing stakeholder intentions. This chapter develops a computational framework for translating these forces into spatial data. Site conditions are discretised into a 10 × 10 m coordinate grid, while stakeholder priorities identified through OODA modelling are encoded as tabular parameters. Using the Waddington Landscape as a conceptual framework, these datasets are translated into parallel fields of attraction and constraint, whose intersection reveals spatial bifurcations and candidate areas for architectural intervention.

Larung Gar has developed through a continuous negotiation between self-organised religious life and external regulation. Its dense fabric emerged incrementally through informal construction, while successive interventions have introduced new spatial, infrastructural and safety constraints. Rather than treating this instability as a condition to be resolved, the project examines how architecture and computational systems might operate within it—mediating between cultural continuity, technological change and evolving forms of governance.

Large-scale intervention has repeatedly reshaped Larung Gar's self-built fabric. Safety concerns, infrastructure upgrades and regulatory measures have introduced clearer circulation routes and more formalised construction, transforming a settlement originally produced through incremental and largely self-organised growth.

This composite drawing brings together the key spatial layers shaping Larung Gar: topography, sacred and public institutions, self-constructed dwellings, and circulation networks. The valley terrain establishes the primary spatial framework, naturally concentrating development along the lower slopes and valley floor. Within this landscape, major religious and public buildings act as stable institutional anchors, while dense self-built dwellings spread incrementally along the contours. Over time, vehicular infrastructure and later pedestrian corridors introduced through planning interventions have further structured the settlement. Together, these layers reveal an urban form shaped through the ongoing interaction between self-organised construction and regulatory intervention.

The model is used to identify and compare the spatial intentions of different stakeholders, translating their priorities, constraints and feedback loops into parameters for the subsequent spatial simulation.

Kenpo | *Student Practitioners* | *Government* | *External Actors*

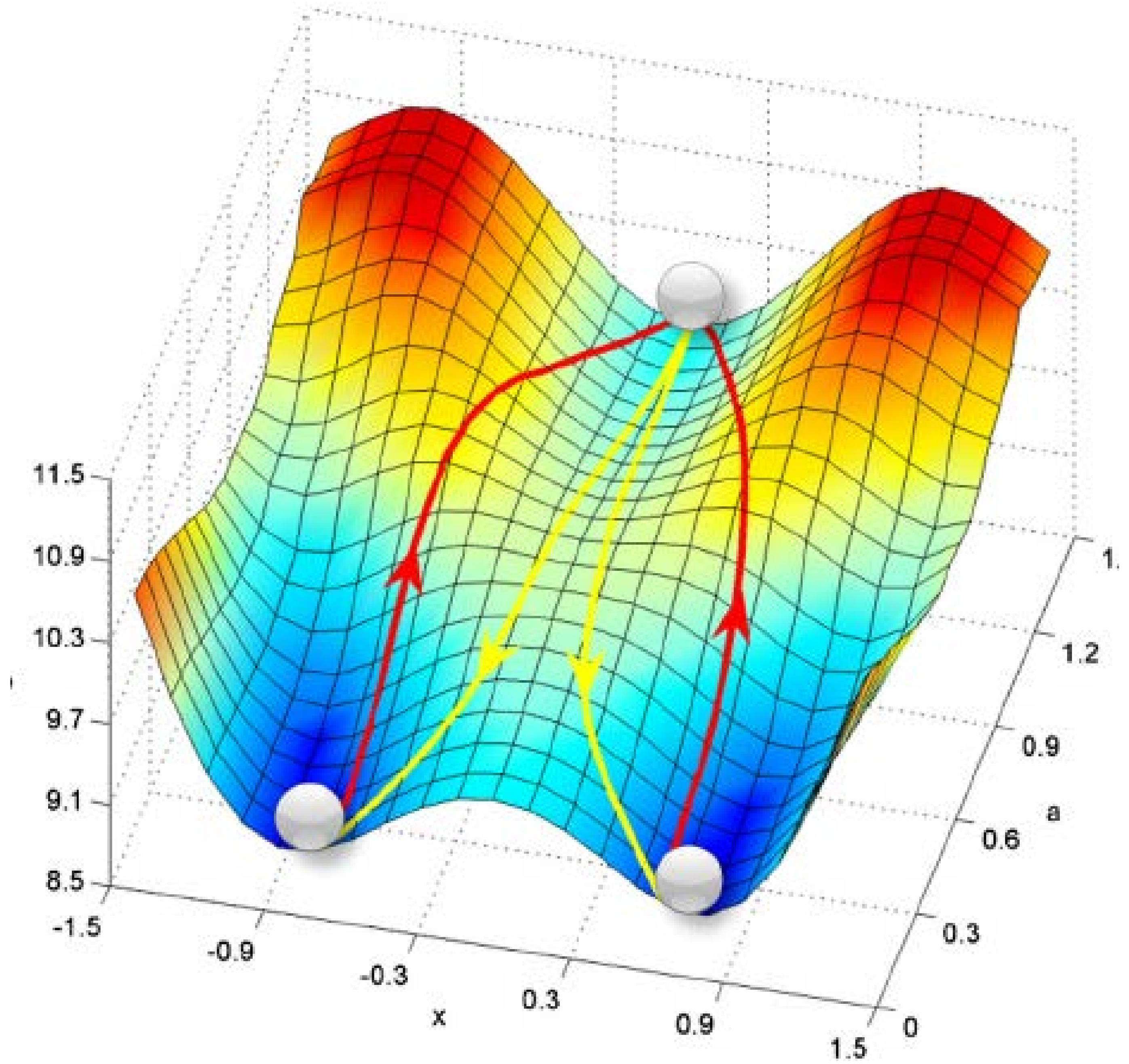
The Kenpo is a senior monastic teacher who combines pedagogical, religious and institutional authority. Beyond guiding Buddhist education and spiritual practice, the Kenpo plays a central role in decisions concerning the academy's development and the long-term continuity of the monastic community.

Students are drawn by religious study, community life, and the possibility of remaining within Larung Gar. Their continued arrival sustains the academy while also contributing to spatial congestion and informal growth.

Intervention is primarily driven by concerns over social stability, regulation and public pressure. These forces periodically translate into planning adjustments, legal education and large-scale spatial restructuring.

Media, NGOs and international observers influence how interventions are publicly interpreted. Through coverage and critique, they introduce an external feedback mechanism that affects the broader narrative surrounding governance.

1.1 Spatial Simulation

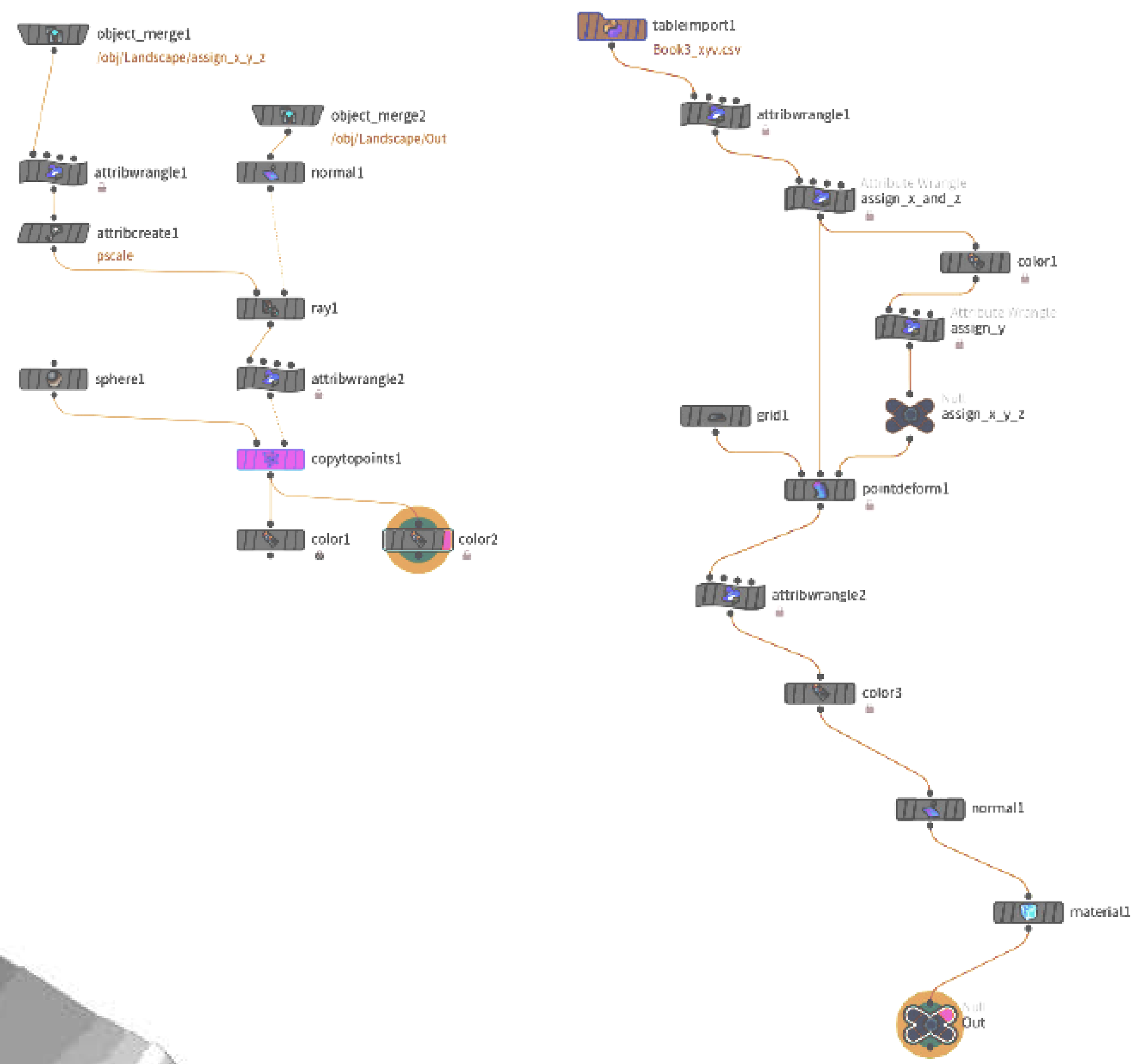


The Waddington Landscape is adopted as a computational metaphor for modelling competing spatial intentions. Stakeholder priorities identified through the OODA analysis are translated into attractors, resistance fields and boundary conditions, allowing possible spatial trajectories to be visualised as a continuous decision landscape.

Spatial Discretisation & Data Encoding

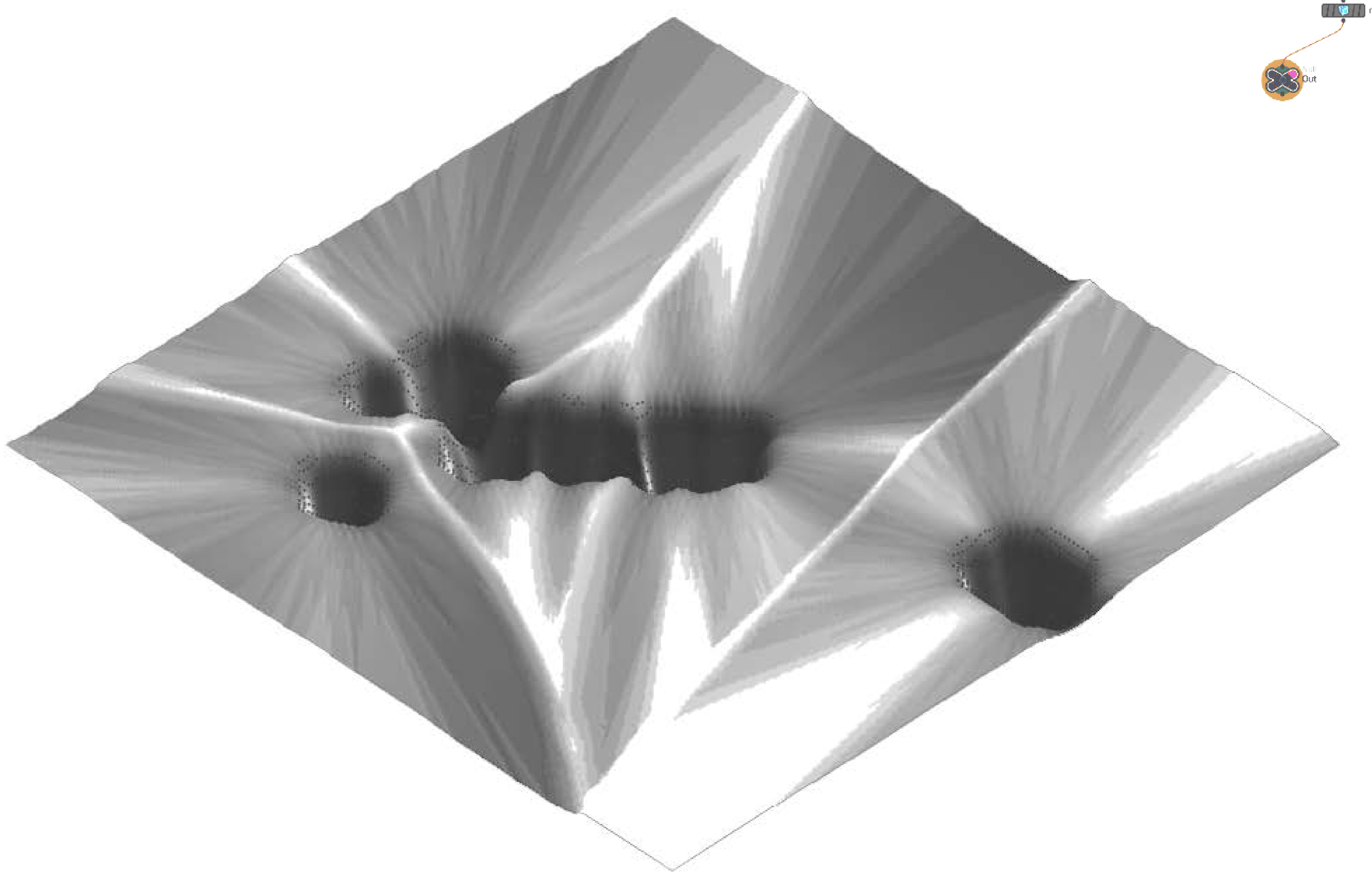
The selected area of Larung Gar was rasterised into a regular 10 × 10 m grid, with each cell assigned a unique coordinate. This transformed the continuous site into a machine-readable spatial dataset, allowing each cell to be recorded as a row within a tabular data structure. Spatial conditions, stakeholder intentions and other parameters could therefore be systematically entered, compared and processed across the site, forming the computational basis for the subsequent attraction and resistance simulations.

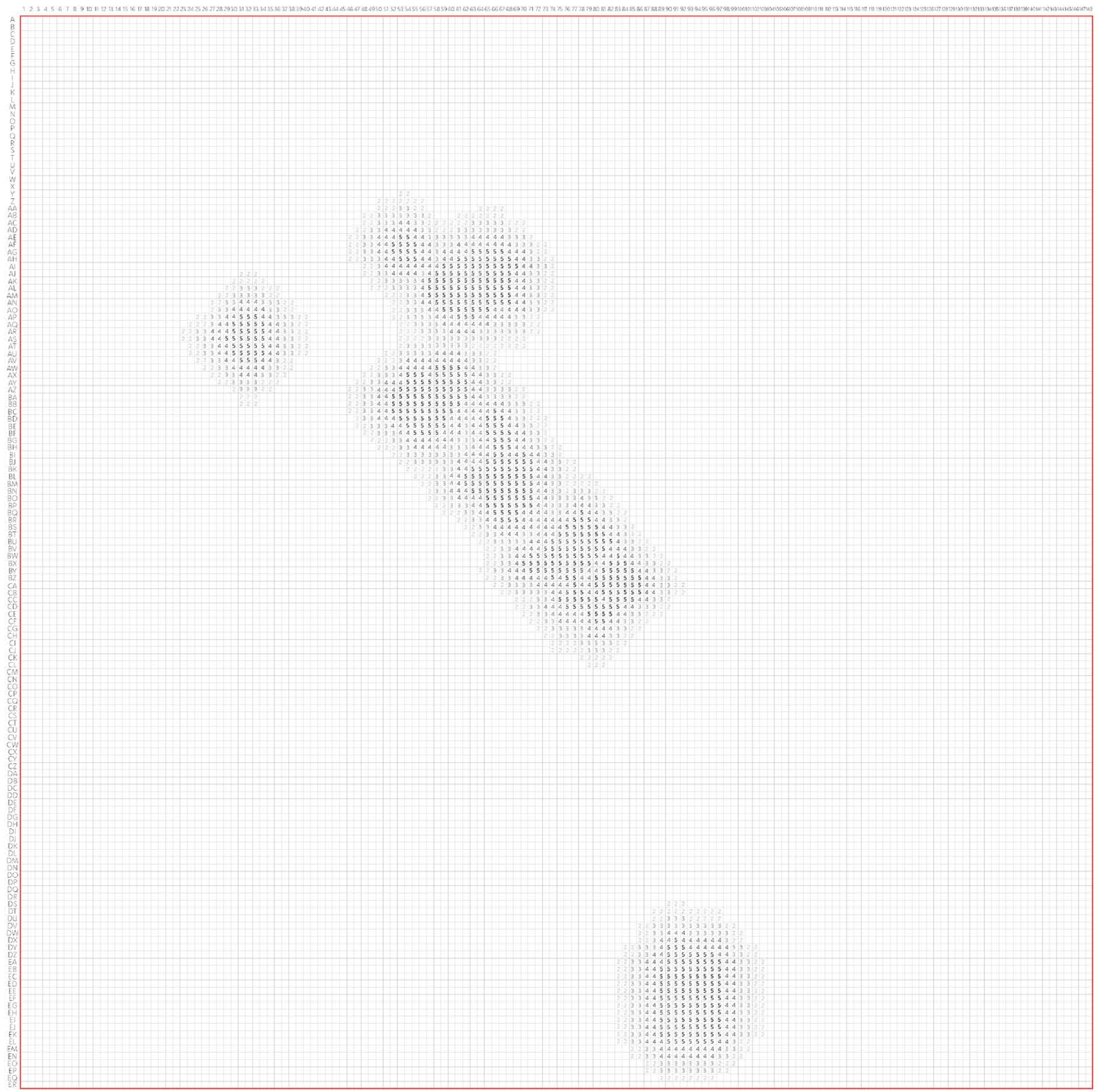
To model the spatial tendency toward religious centres, the site grid was populated with weighted coordinate points representing the relative attraction generated by major monastic institutions. Each point was assigned an intensity value and recorded within a tabular dataset, allowing spatial attraction to be encoded numerically across the site. This structured data formed the basis for the subsequent generation of the attraction field.



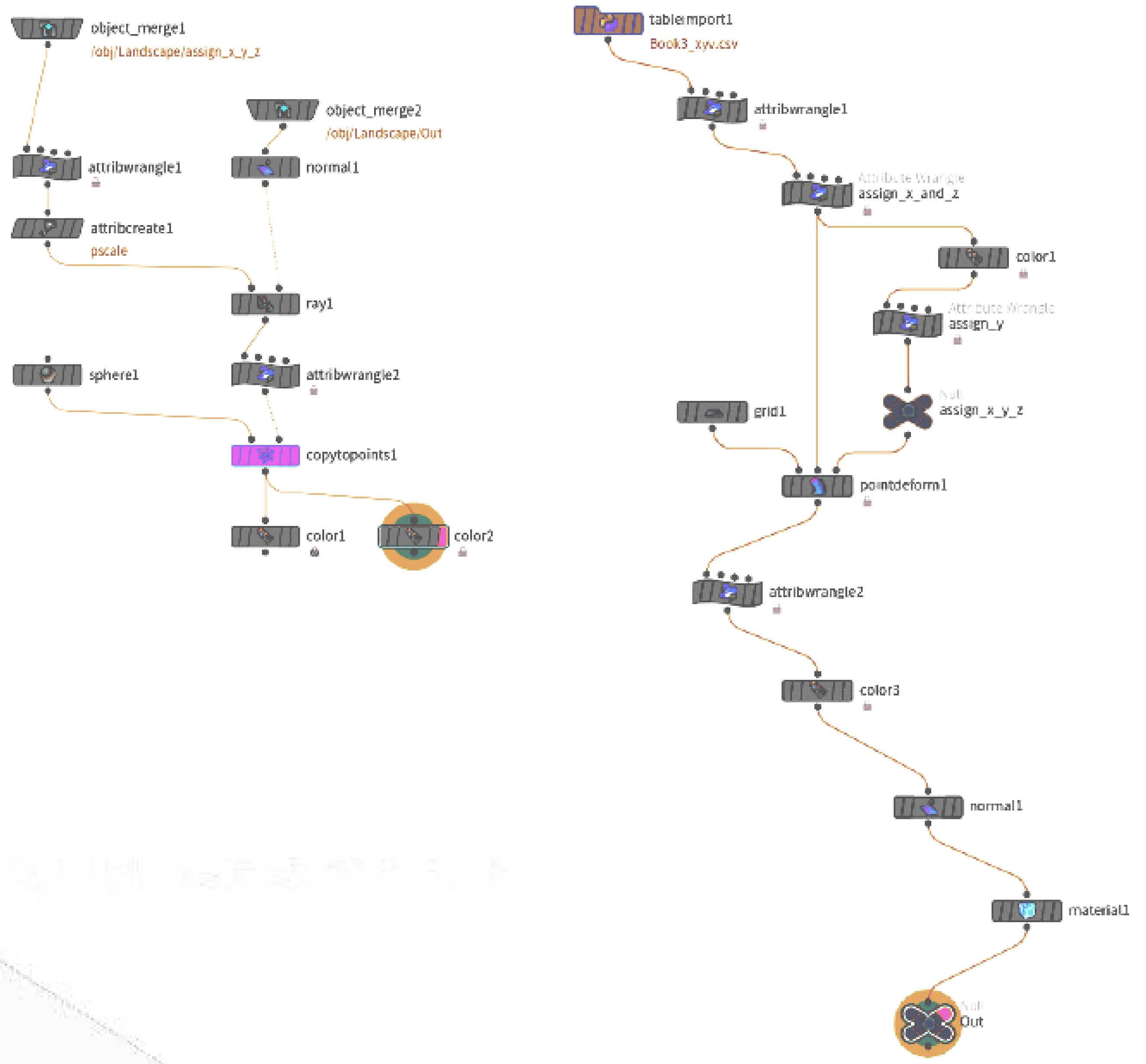
Religious Attraction Landscape

Valley *— spatial attraction*
Deep basin *— strong religious attractor*
Slope *— directional pull*
Flat field *— limited religious influence*



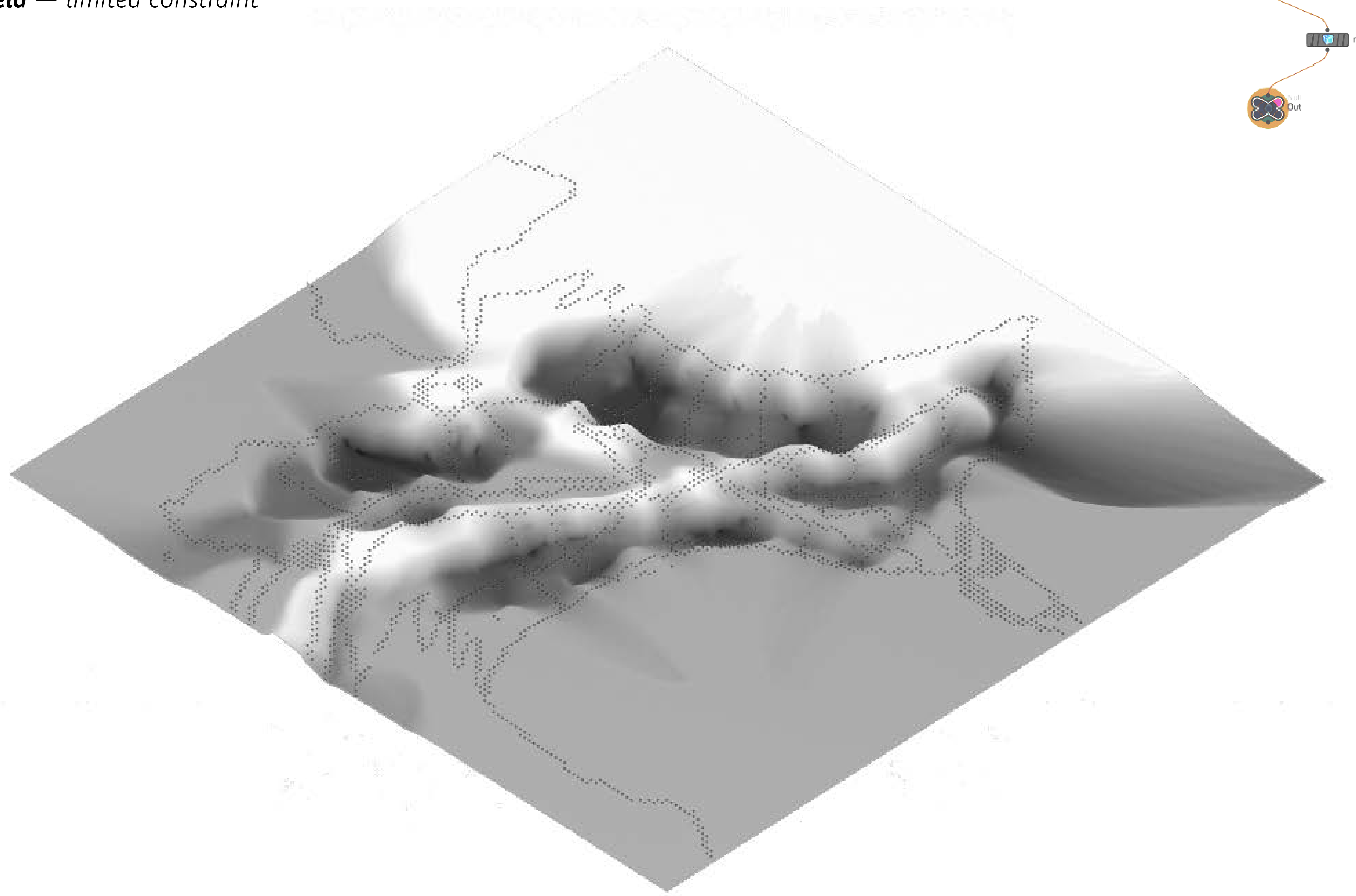


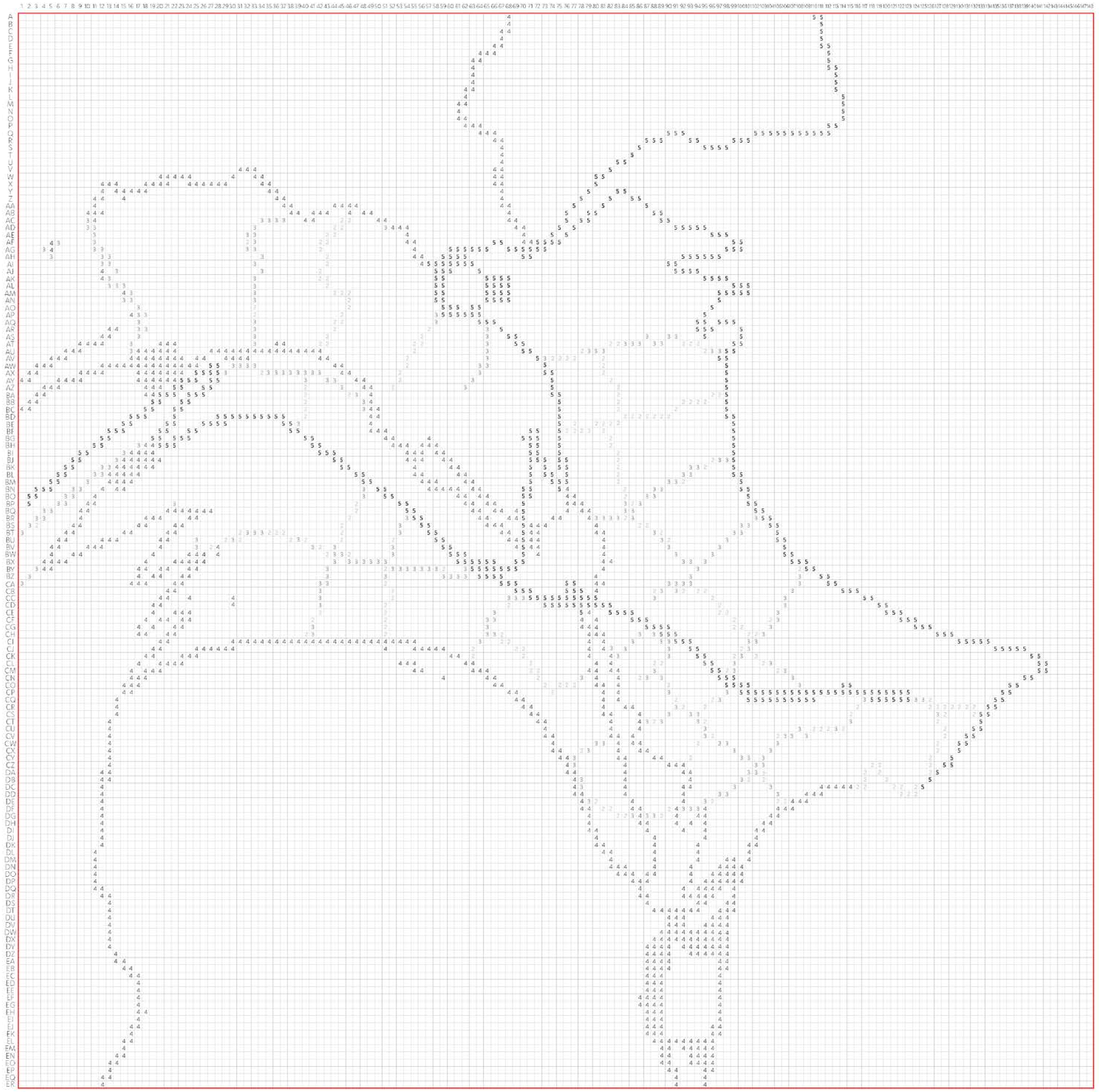
*In parallel, perceived boundaries and spatial constraints were encoded as a second set of weighted coordinates representing constraining forces rather than attraction. These tabulated values were imported into **Houdini**, where a procedural workflow translated point-based heat values into scalar fields and interpolated them into a continuous constraint landscape. The resulting ridges divide the landscape into distinct spatial trajectories, representing conditions that constrain, redirect or separate potential patterns of development.*

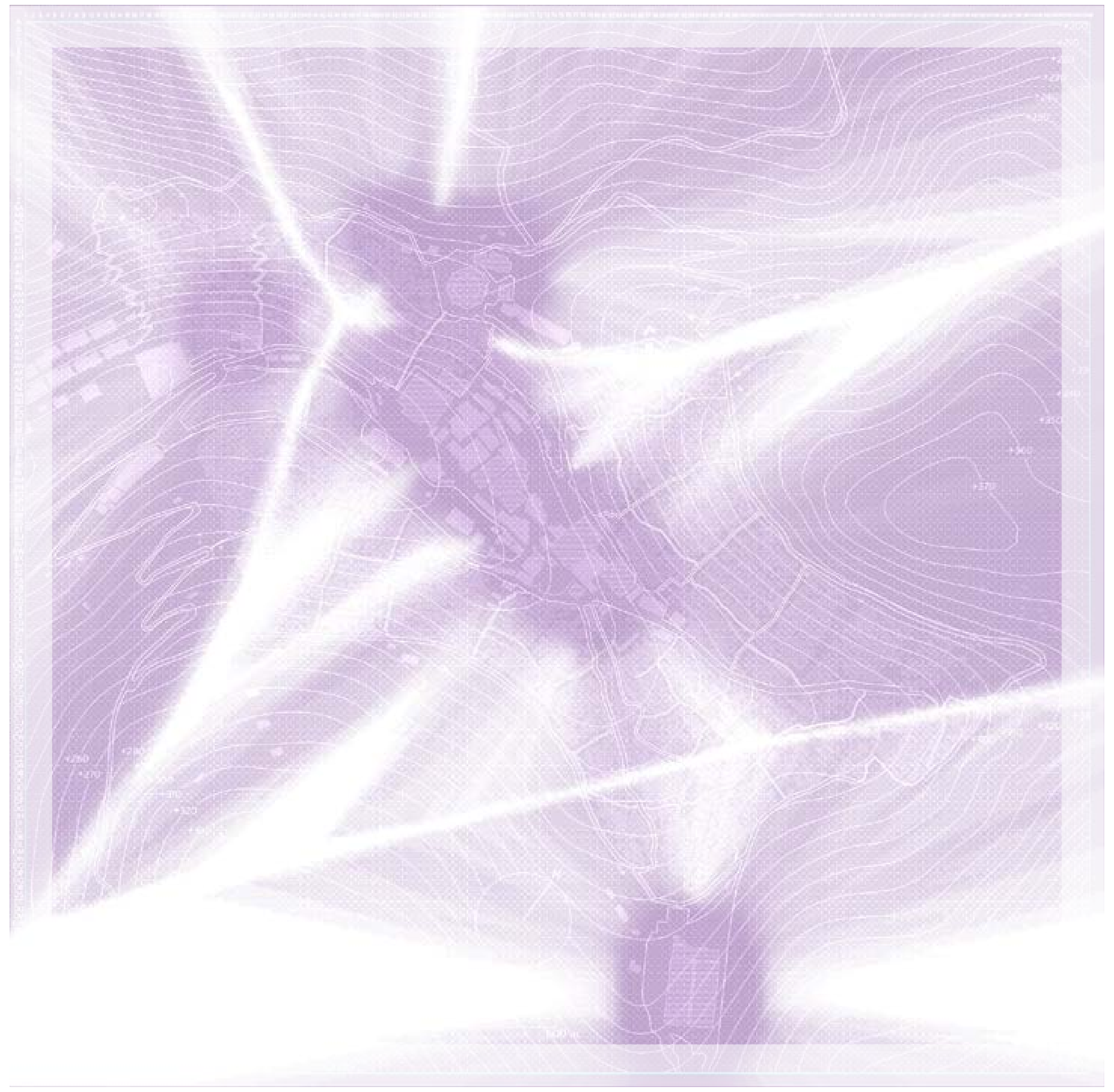


Spatial Constraint Landscape

Ridge *— spatial constraint / separation*
High ridge *— strong boundary or restriction*
Slope *— increasing constraint*
Flat field *— limited constraint*







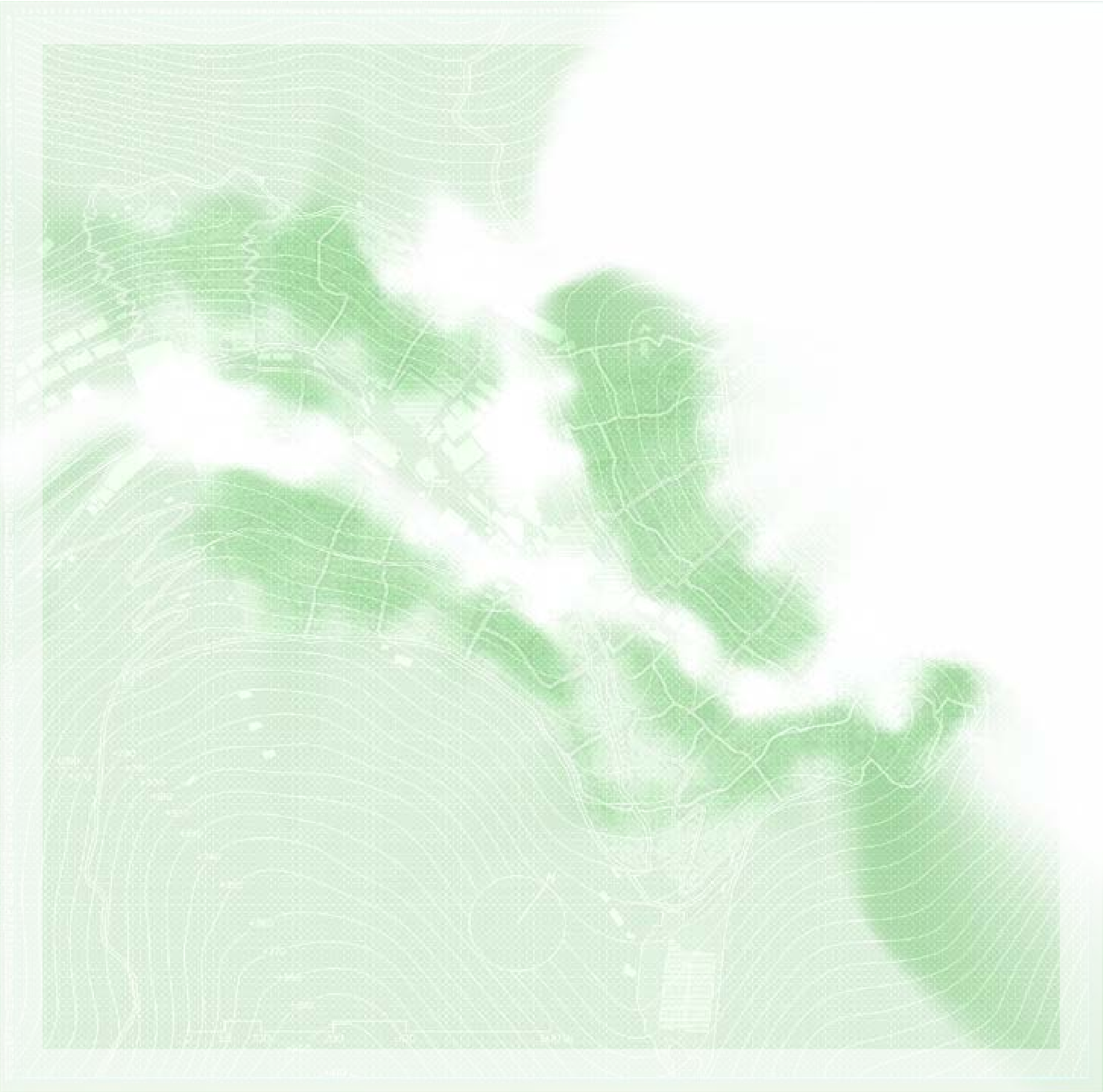
The two simulated landscapes—religious attraction and spatial constraint—are superimposed to produce a composite decision field representing competing spatial intentions. The purple field represents attraction generated by religious centres, while the green field represents spatial constraints produced by boundaries, infrastructural conditions and regulatory forces.

Regions of interest to **Both side**

Areas of interest only to **Buddhists**

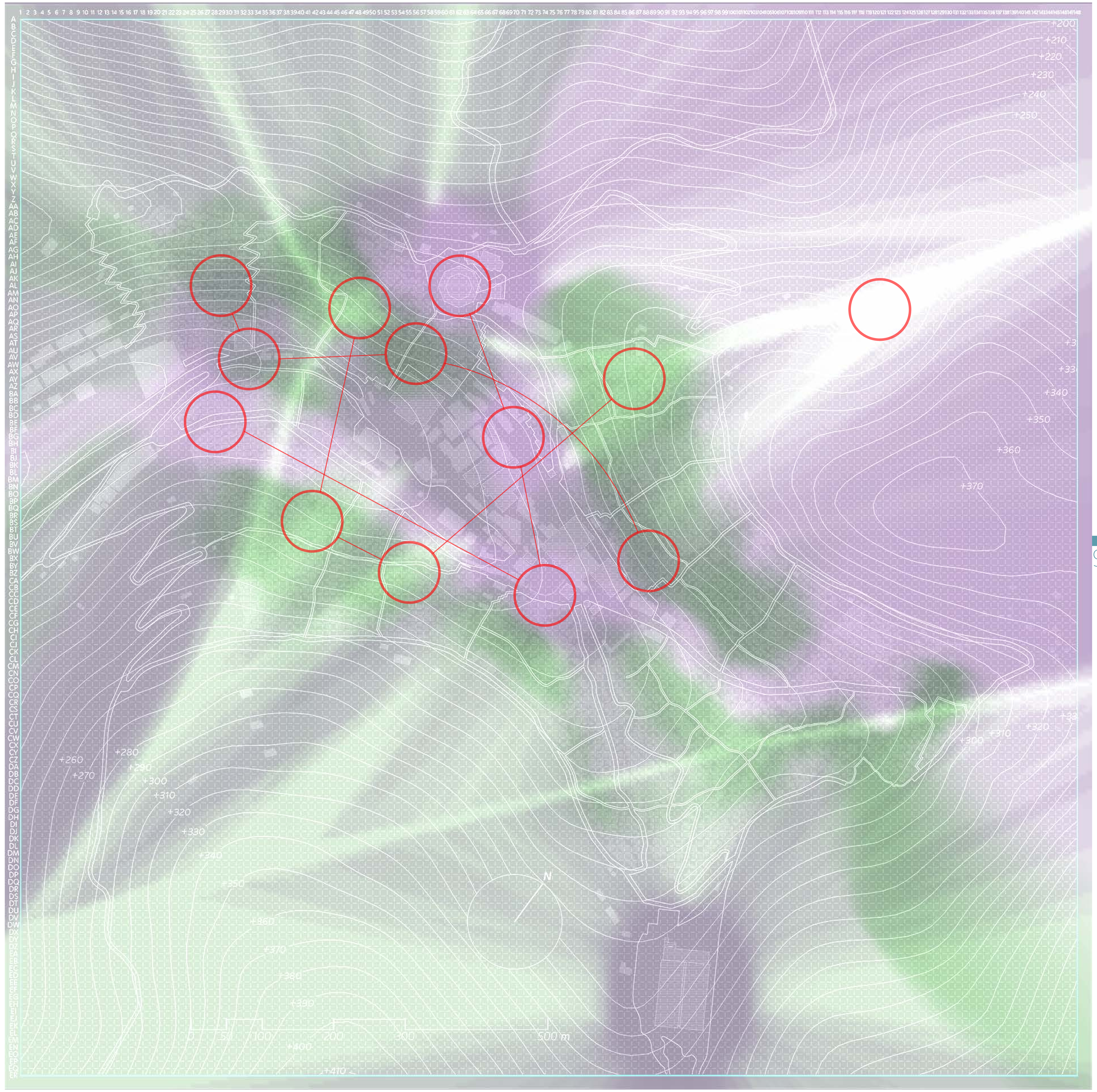
Regions preferred only by **administrative forces**

Regions of interest to **neither side**



Within the composite landscape, valleys indicate spatial attraction while ridges represent constraining or separating forces. Where attractive valleys encounter constraining ridges, trajectories are redirected, divided or intensified, producing zones of spatial bifurcation.

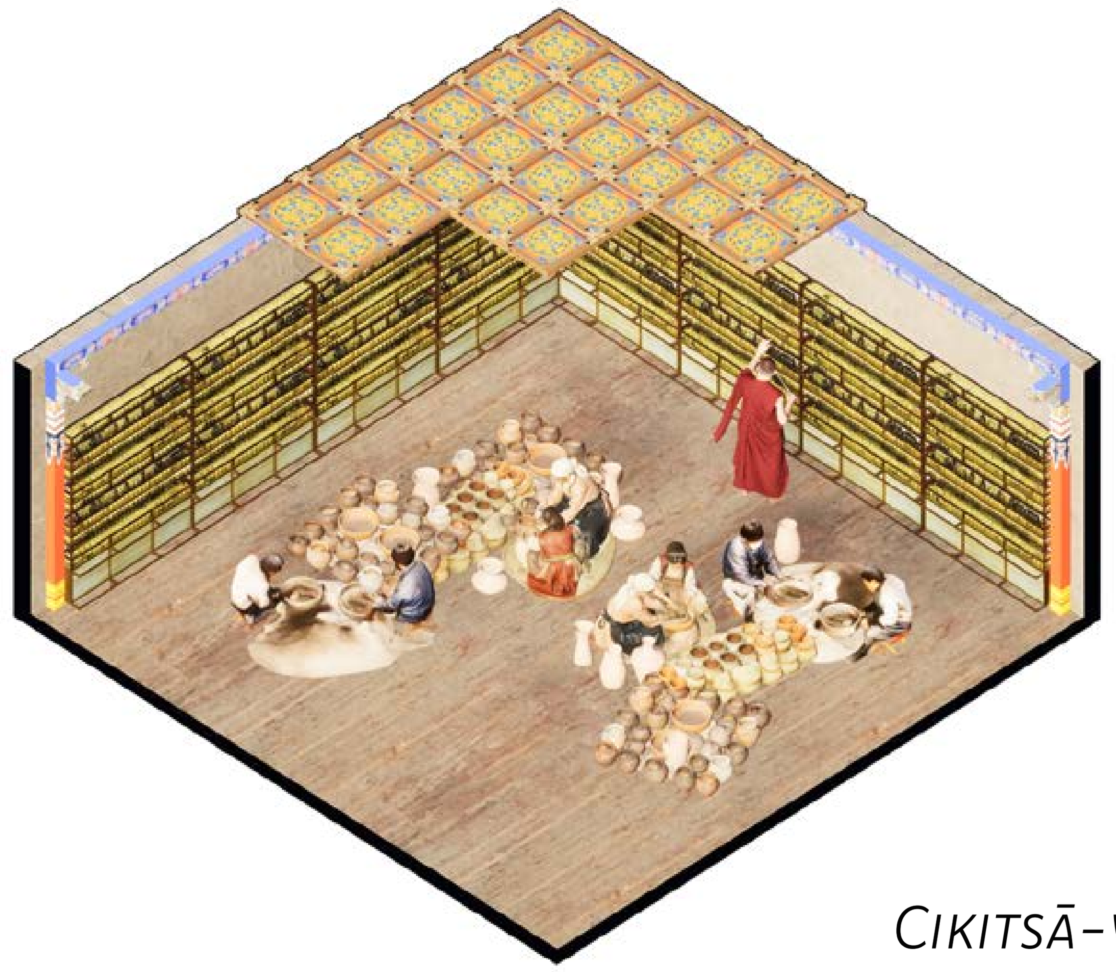
The analysis focuses on bifurcation zones where competing stakeholder intentions intersect. These locations represent decision points at which alternative spatial trajectories become possible, and are therefore identified as candidate areas for architectural intervention.



1.2 Digital Preservation of Cultural Practice

Five Sciences
Mandala Taxonomy
Machine-Readable Encoding

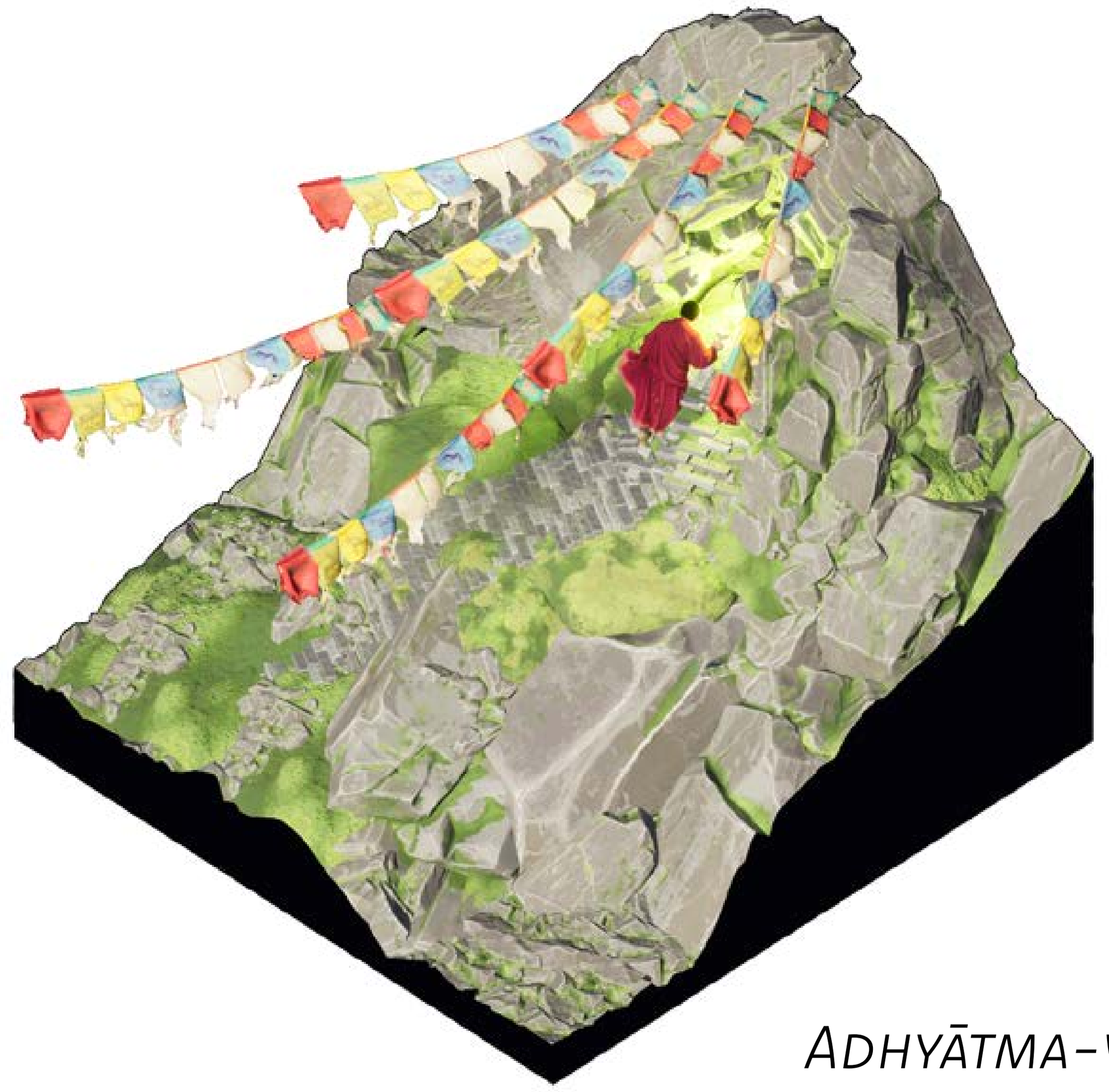
This chapter explores the digital preservation of cultural practice through the "Five Sciences", a framework for understanding Tibetan Buddhist knowledge and practice. From this broader system, the sand mandala is selected as a specific knowledge structure for further study. Its spatial organisation, compositional logic and making processes are analysed, decomposed and classified before being translated into machine-readable forms. Together, these studies examine preservation at the level of cultural knowledge and practice.



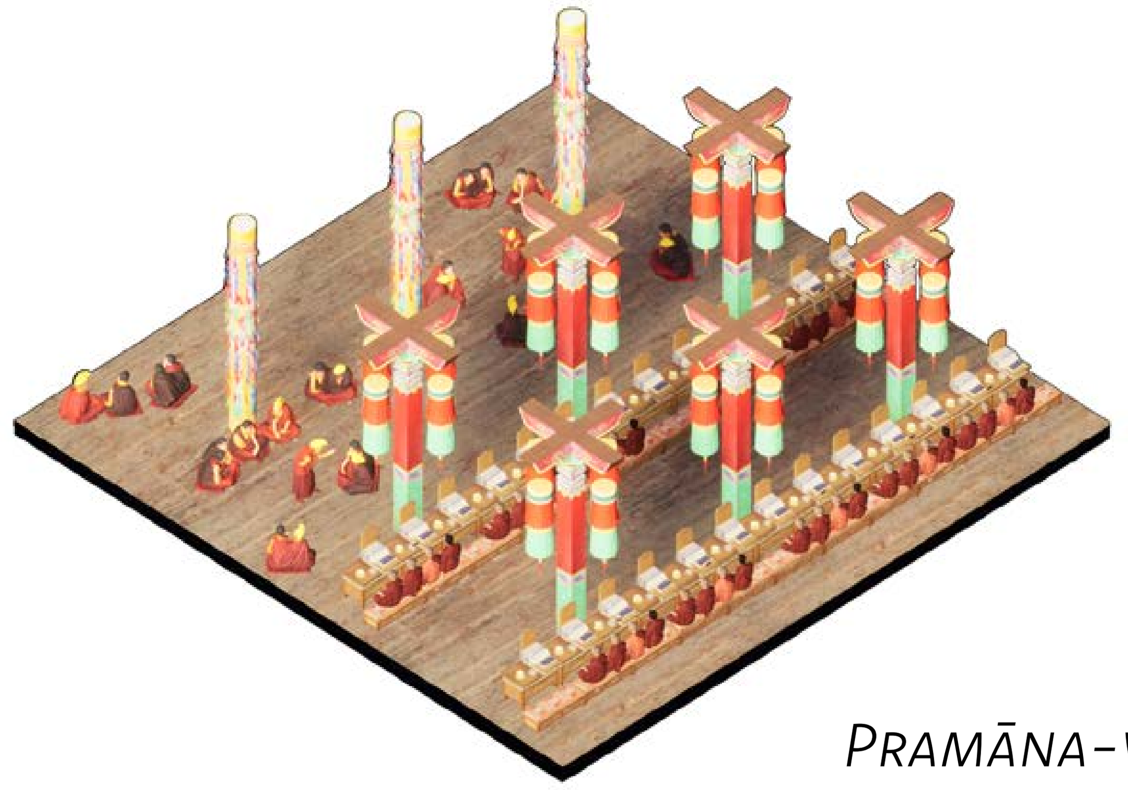
Cikitsā-vidyā
"Science of Medicine"

Śabda-vidyā
"Science of Language"

Śilpa-vidyā
"Science of Arts and Crafts"



Adhyātma-vidyā
"Inner Science"

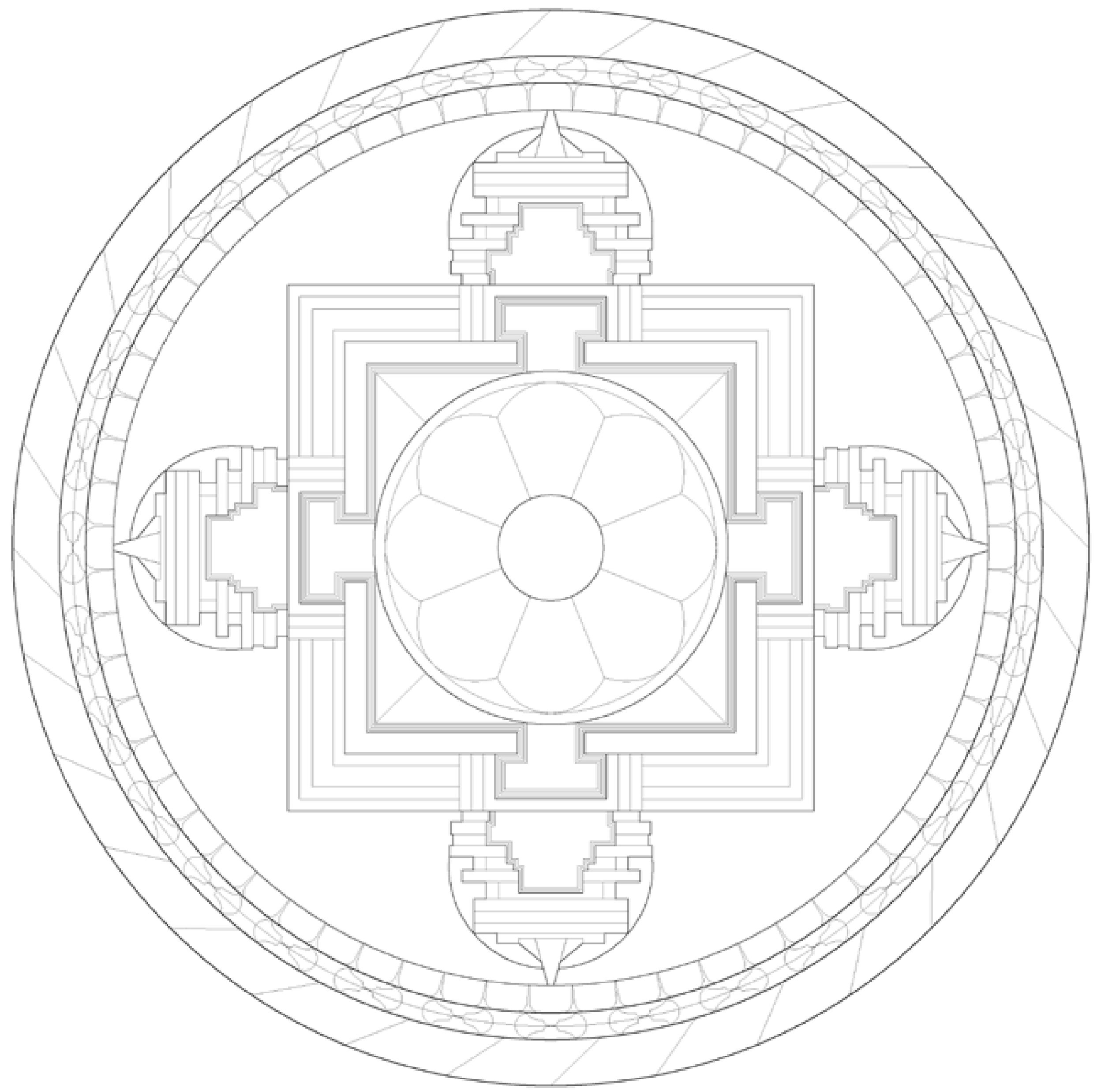


Pramāna-vidyā
"Science of Logic"

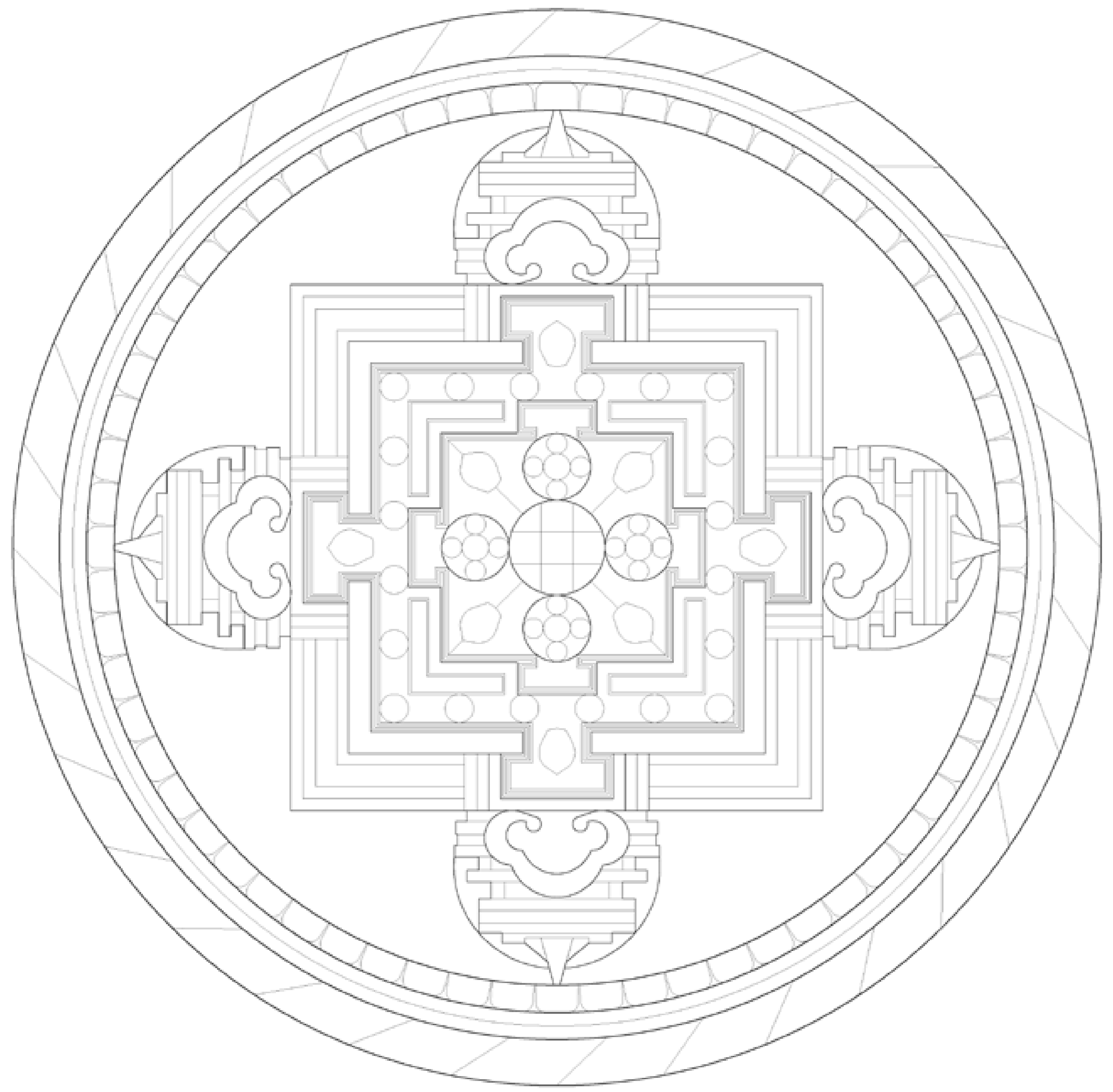
1.2 Mandala Taxonomy

The sand mandala was selected as a specific cultural practice through which knowledge could be studied as a structured system. A comparative analysis of different mandala types was used to identify recurring geometric elements, spatial hierarchies and compositional relationships. Rather than treating each mandala as an isolated image, the study looks for shared organisational principles that can later be classified and translated into a machine-readable structure.

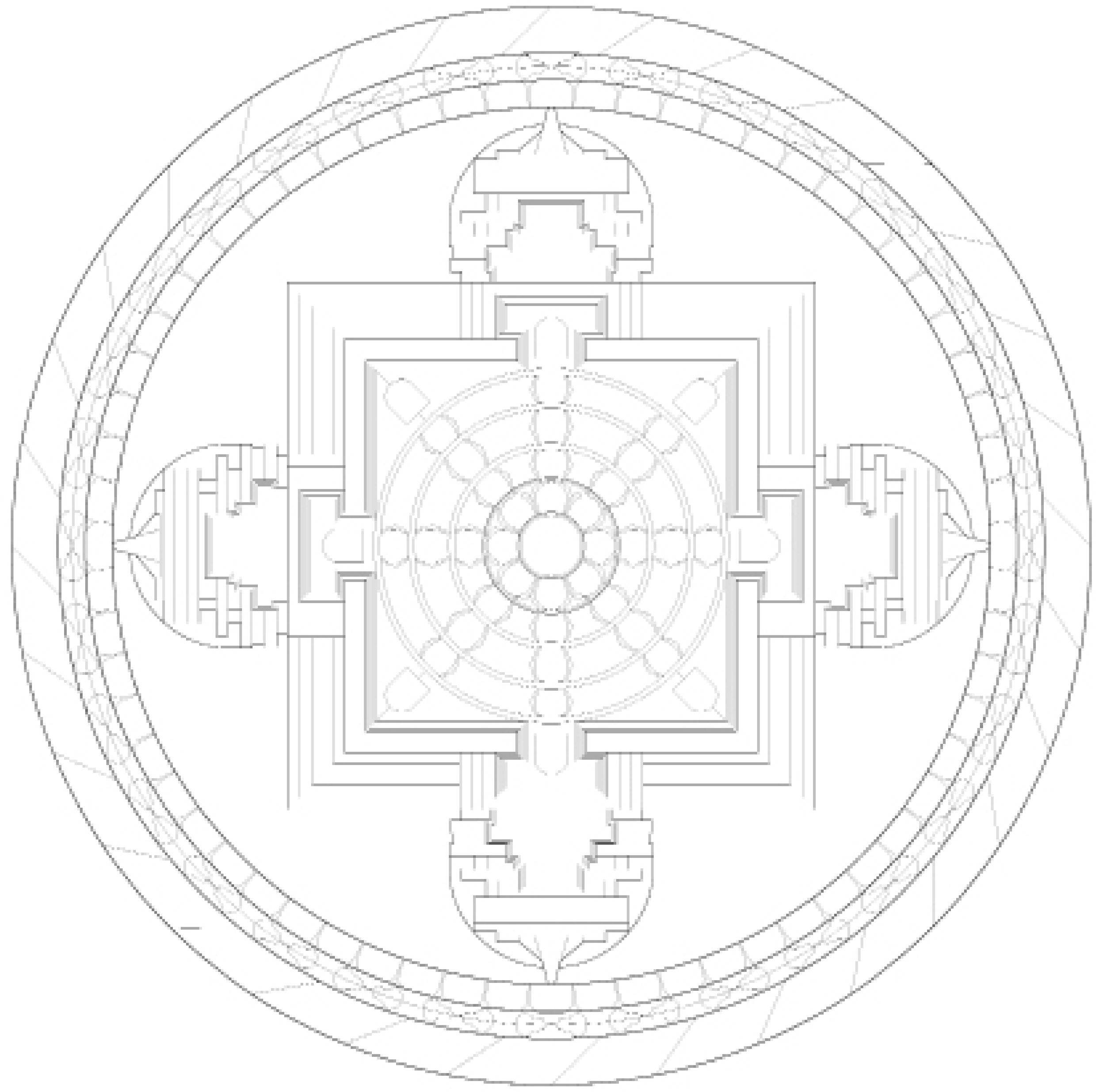
Different mandala compositions were redrawn and compared to distinguish recurring spatial structures from type-specific variations.The recurring components identified through this comparison form the basis of the following taxonomy and encoding process.



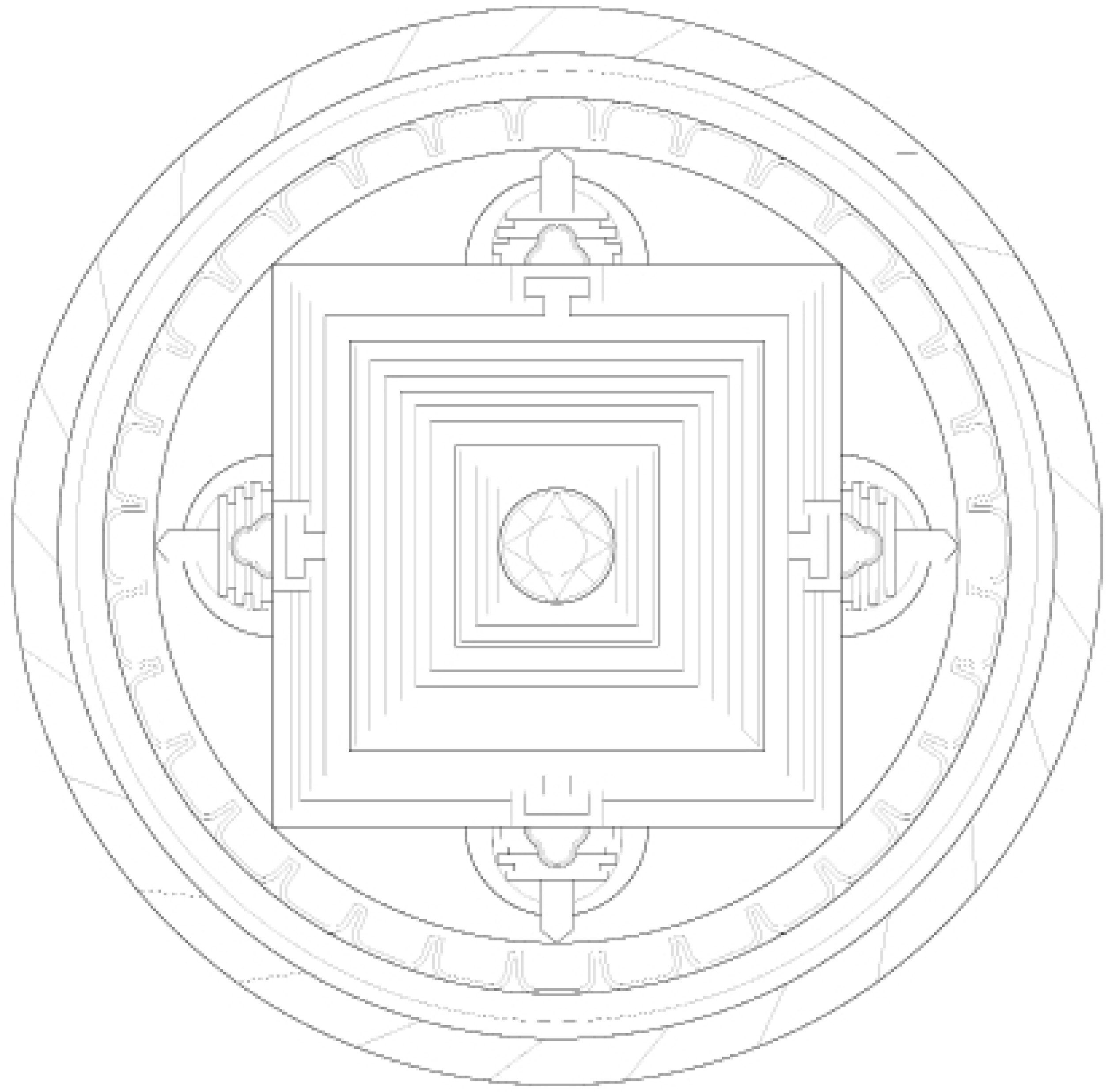
Symbolic-Centred Mandala



Narrative Mandala

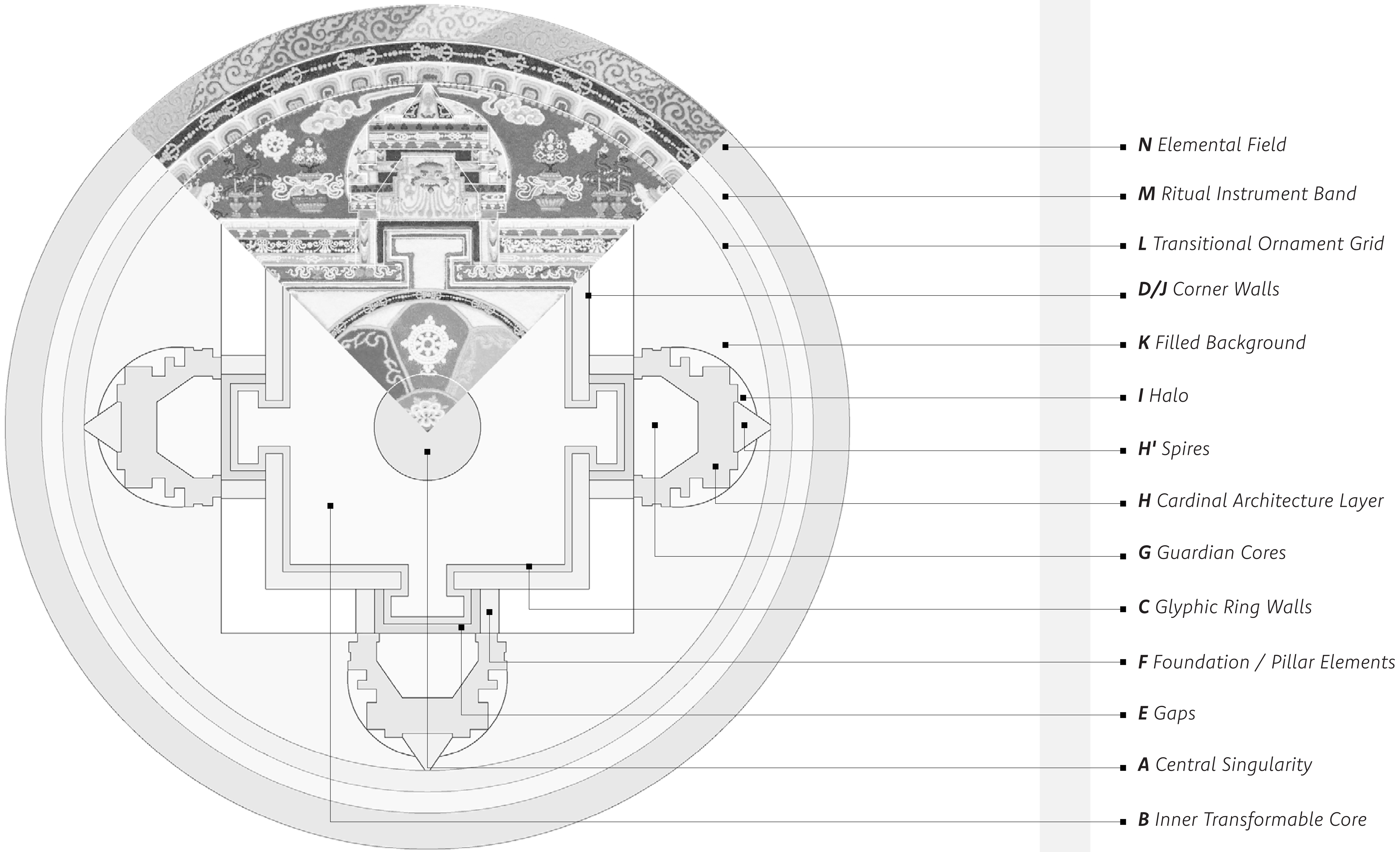


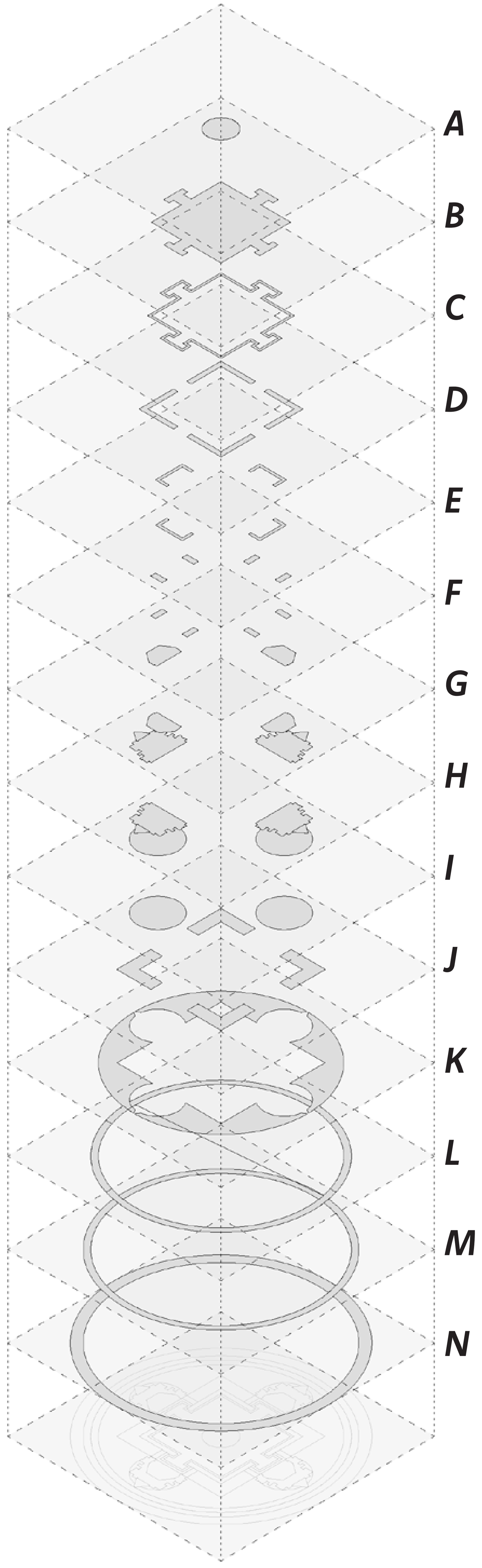
Figural–Cosmological Mandala



Architectural Mandala

Mandala compositions are structured through architectural elements such as centres, boundaries, gateways, walls and layered enclosures, forming a clear spatial hierarchy. By comparing multiple examples, these recurring spatial elements are decomposed and classified into a consistent taxonomy, forming the basis for subsequent machine-readable encoding.





1.3 System Architecture & Implementation

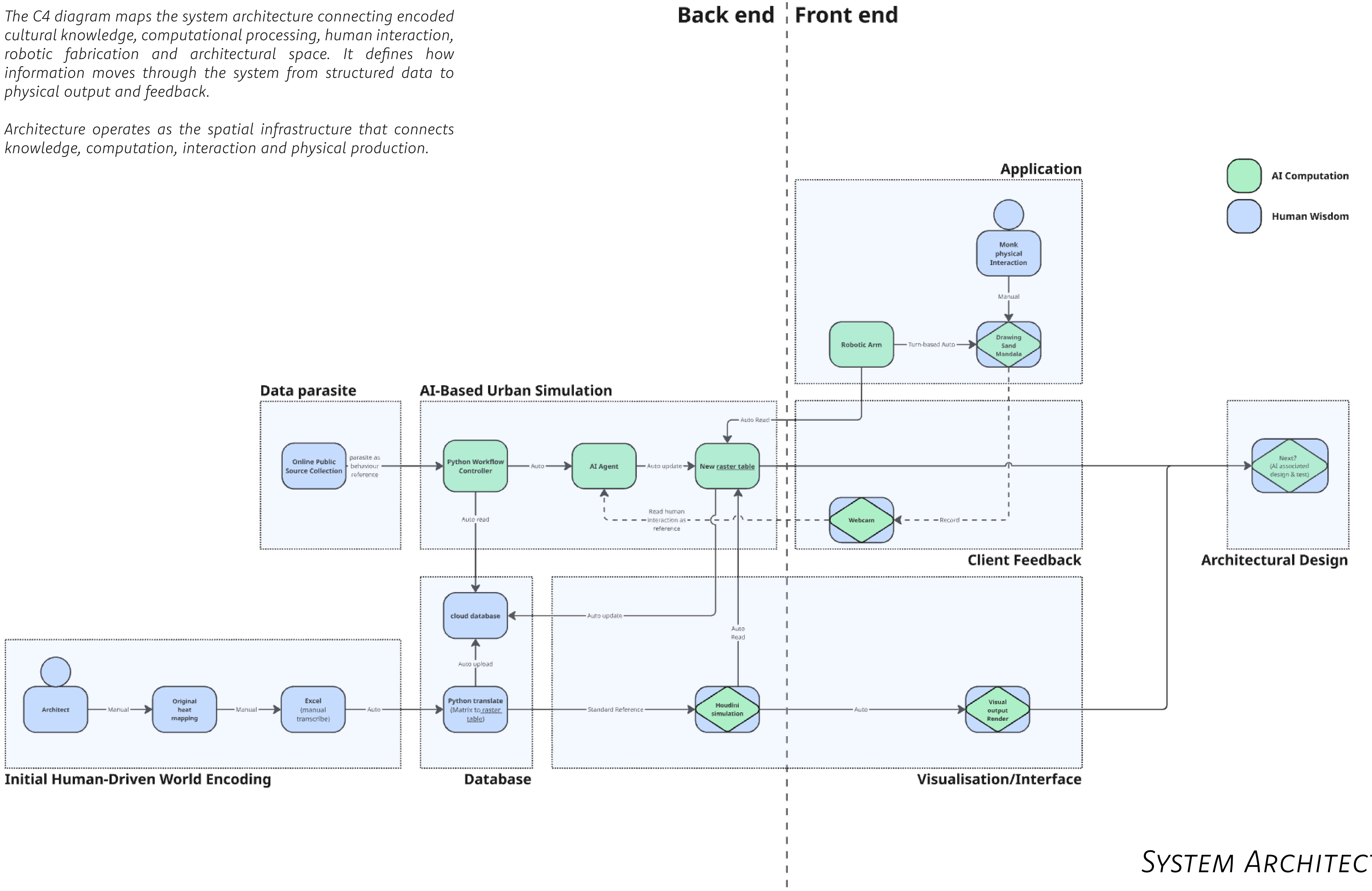
C4 System Architecture
Tool & End-Effector Development
Computational Pipeline
Robotic Integration & Sand Deposition
Architectural System Integration

This chapter asks how cultural knowledge can continue to be shared when direct physical exchange is limited. Building on the machine-readable encoding of the sand mandala, it develops an operational computational and robotic system that translates structured data back into interaction, fabrication and material output.

The following studies examine the system architecture, control pipeline, gesture interface, tool development and robotic sand deposition.

The C4 diagram maps the system architecture connecting encoded cultural knowledge, computational processing, human interaction, robotic fabrication and architectural space. It defines how information moves through the system from structured data to physical output and feedback.

Architecture operates as the spatial infrastructure that connects knowledge, computation, interaction and physical production.

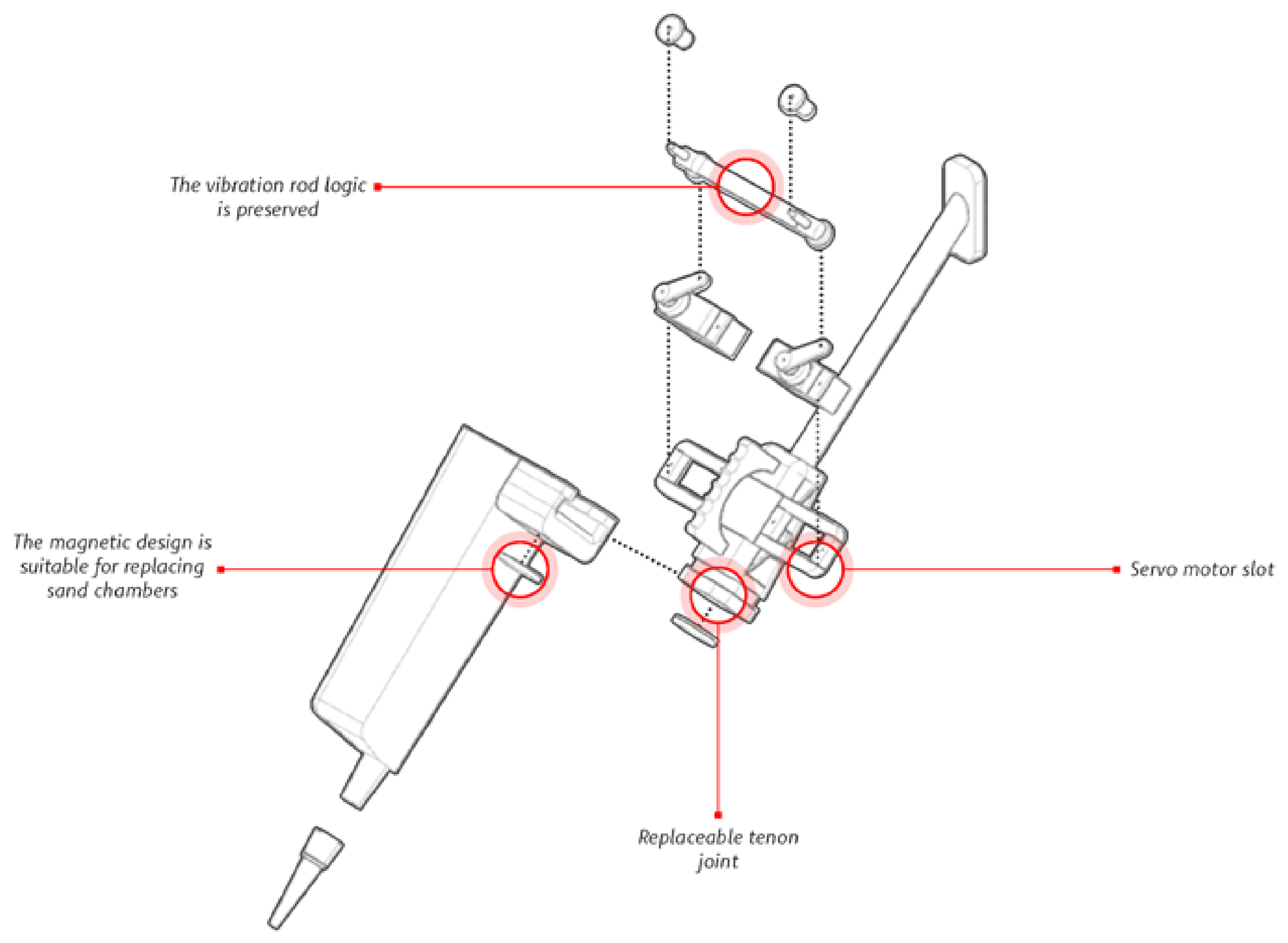


C4
System Architecture

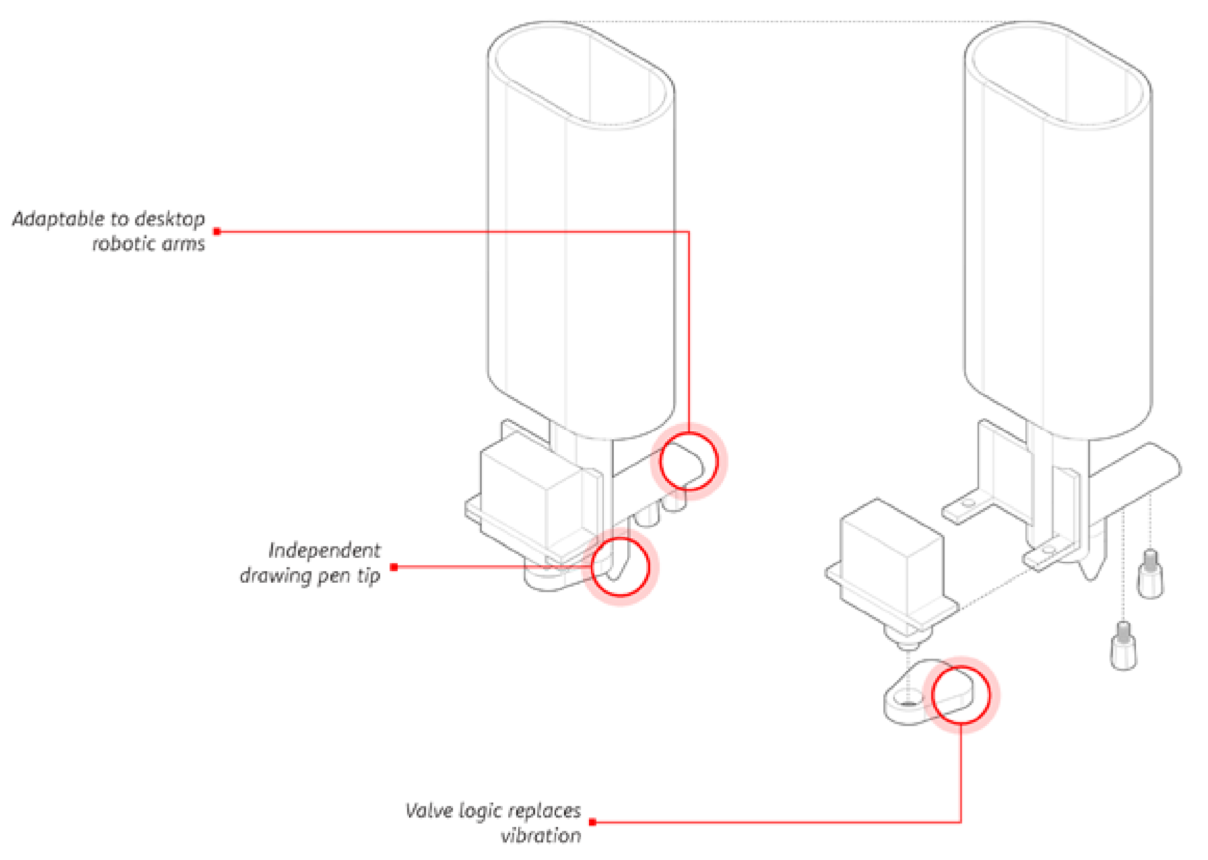
A series of end-effector prototypes explored how sand deposition could be translated into a controllable robotic process. Each iteration was physically built and tested, progressing from vibration-based dispensing to valve-controlled and gravity-fed systems while evaluating material flow, stability and compatibility with robotic movement. The process culminated in an integrated robotic sand deposition head.

Vibration -> Flow Control -> Gravity Feed -> Robotic Integration

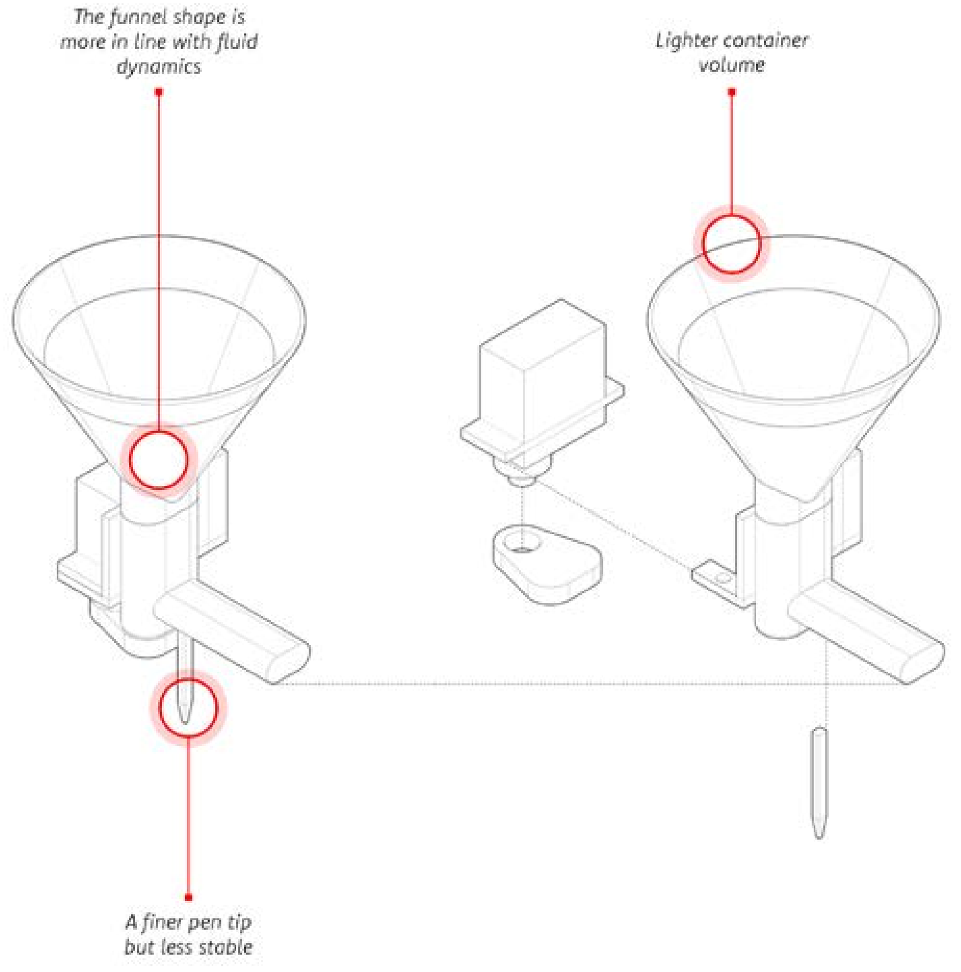
Prototype 3 – Vibratory Sand Dispenser



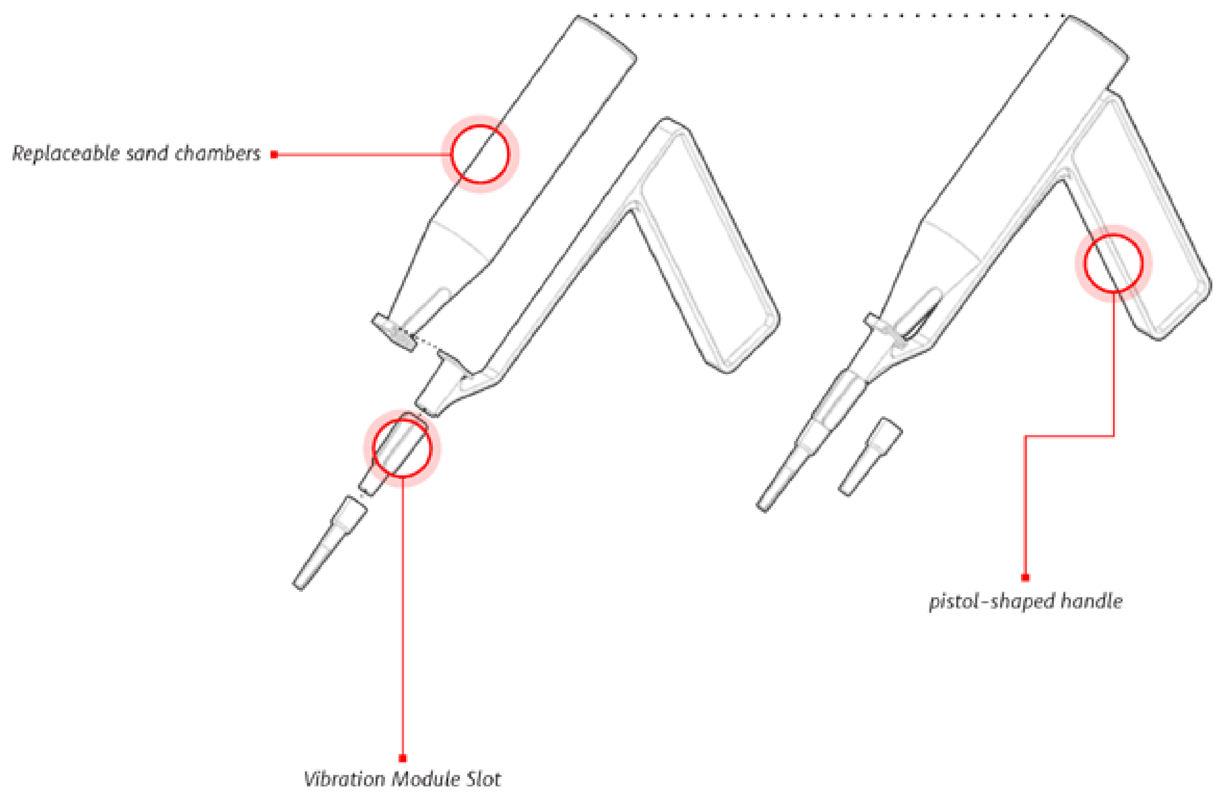
Prototype 6 – Valve-Controlled Sand Dispenser



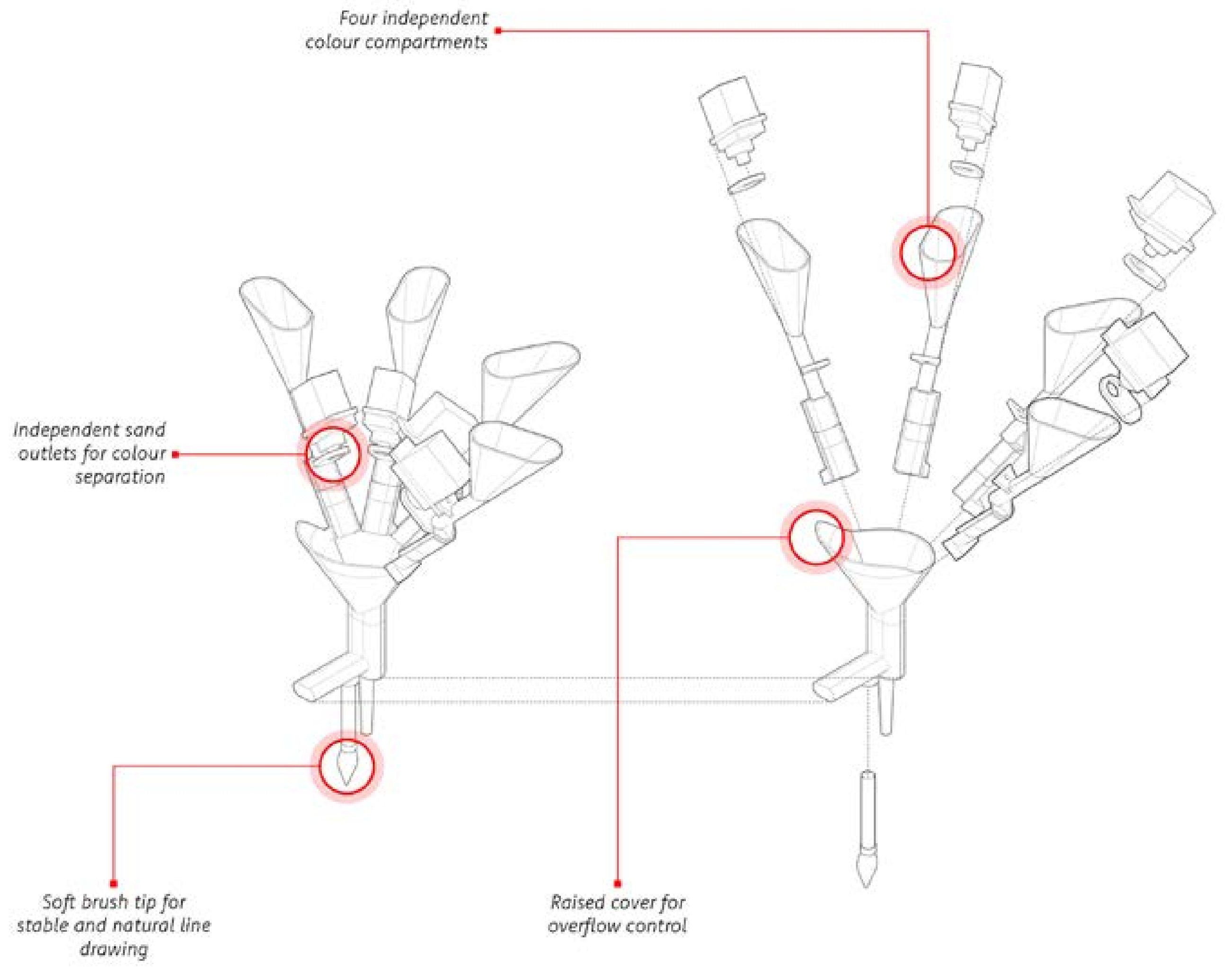
Prototype 7 – Funnel-Based Sand Dispenser



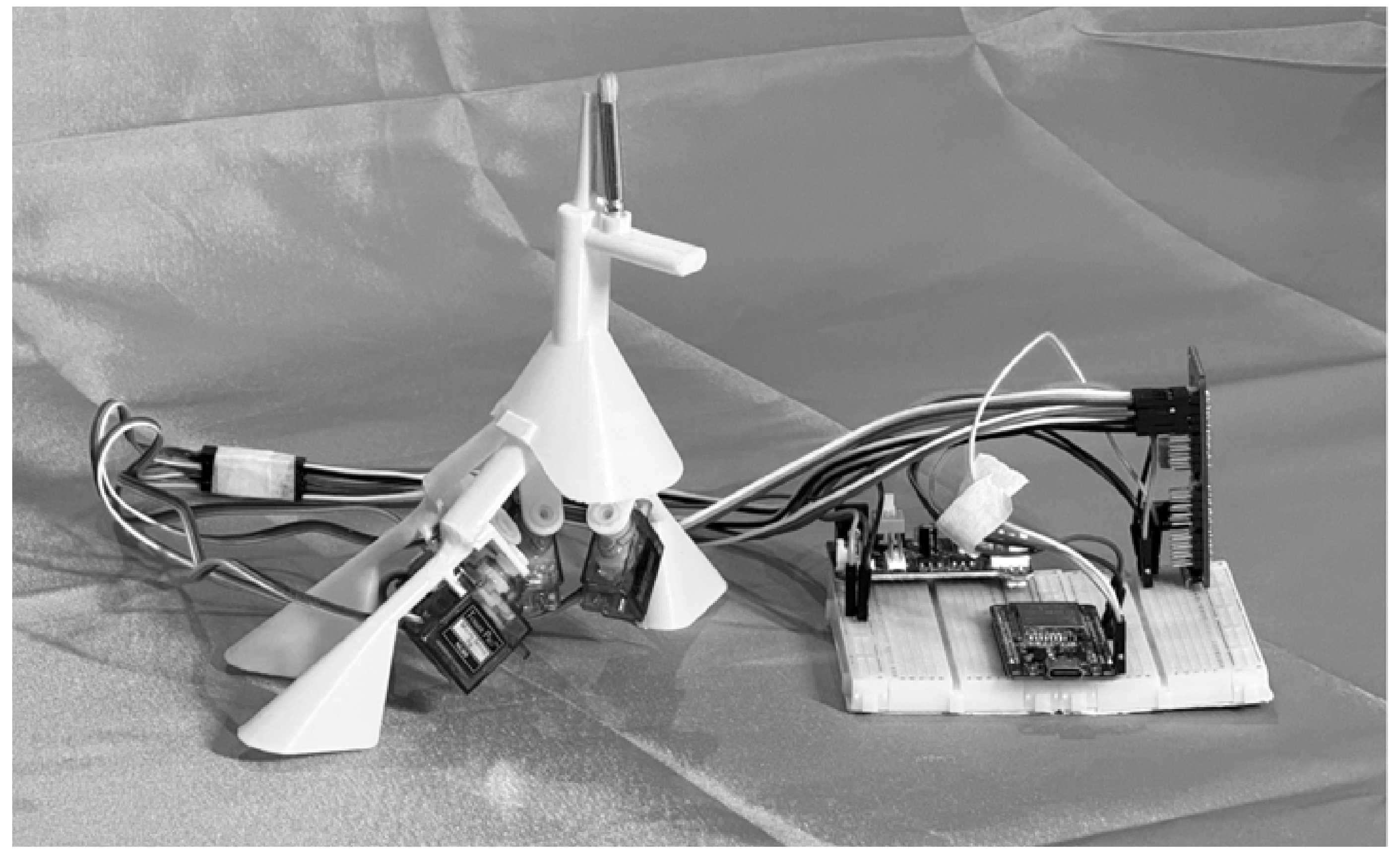
Prototype 4 – Vibratory Sand Gun



Prototype 8 – Robotic Sand Deposition Head



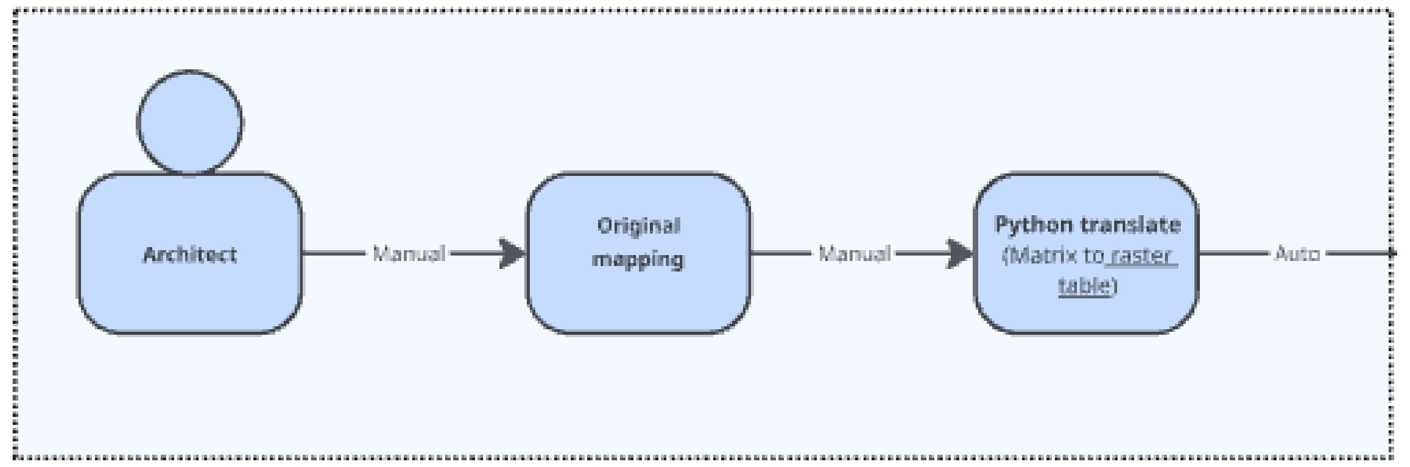
Prototype 8 – Physical Model



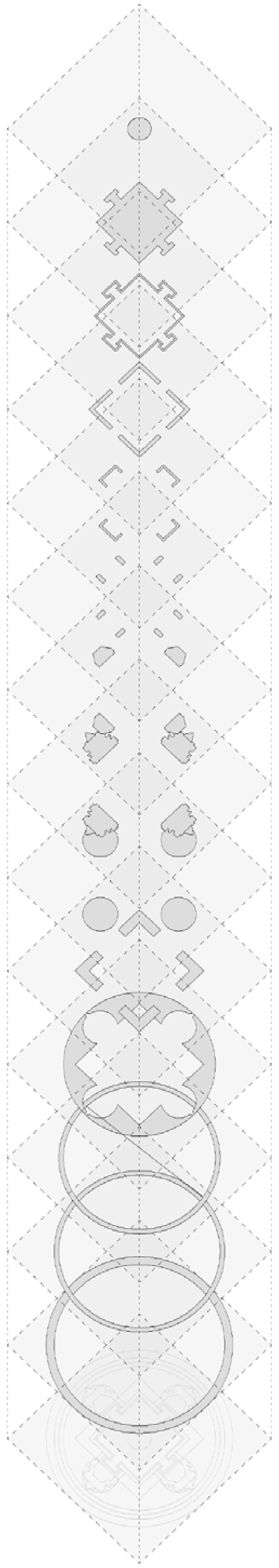
The final deposition head combines four independent sand channels, allowing different colours to be switched during drawing, with controlled outlets and a soft brush tip for stable line formation. Its modular design improves material control and reliability during robotic operation.

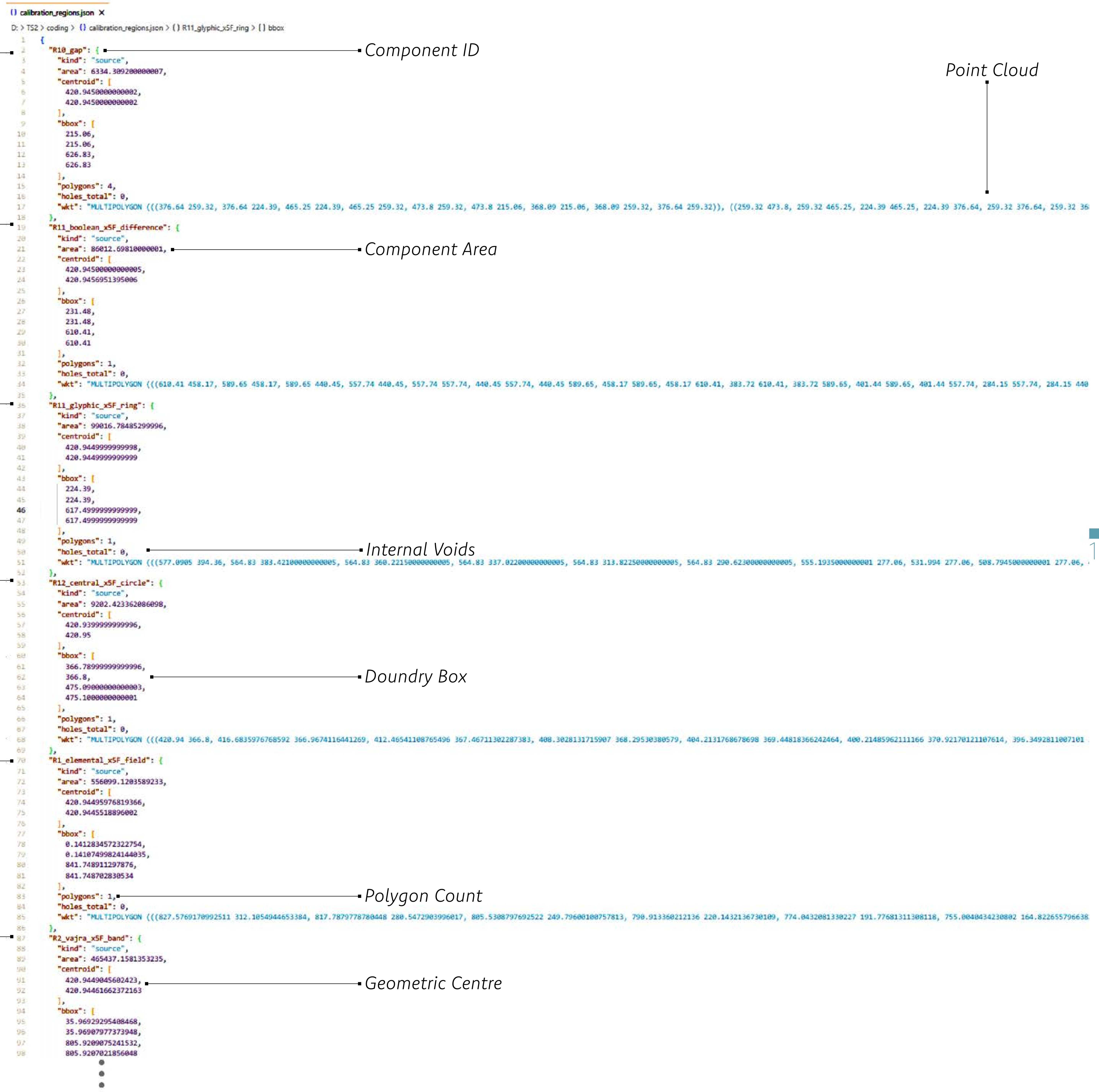
Geometry to Machine-Readable Data

Selected mandala components are manually defined as editable geometry and translated through Python into structured machine-readable data. Bounding regions, point clouds and coordinate sets preserve the geometric relationships of each component while providing the parameters required for subsequent toolpath generation and robotic execution.



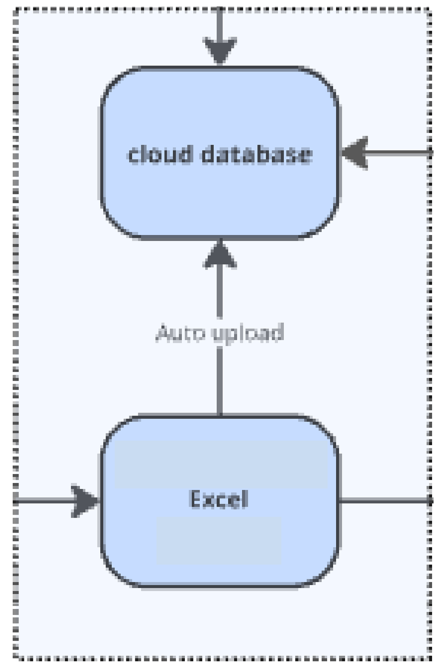
Manual Geometry Input -> Python Translation -> Structured Spatial Data



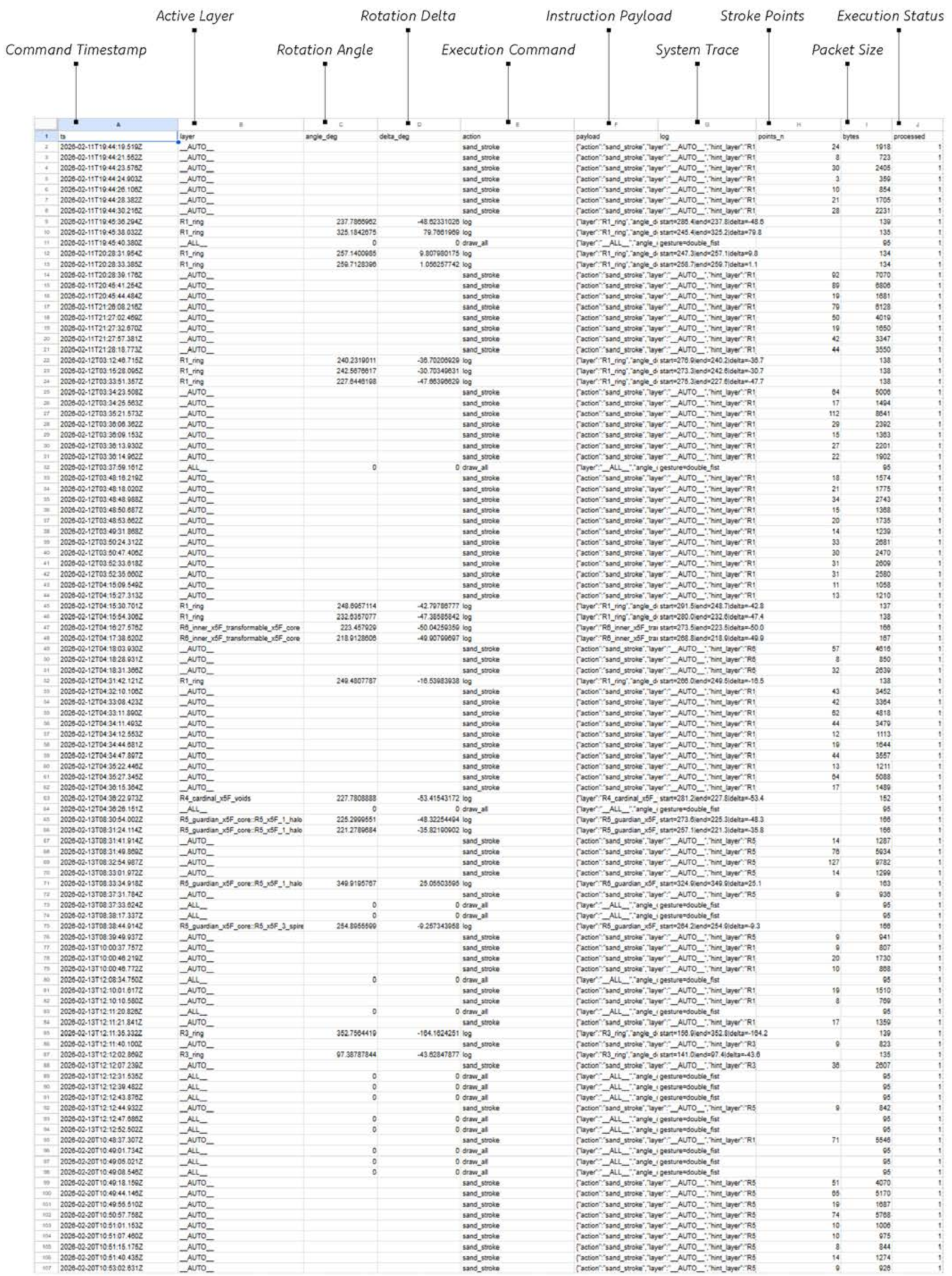


Data Queue & Robot Communication

A lightweight cloud interface was implemented using Google Apps Script and Google Sheets. Apps Script receives JSON requests and manages communication between the computational pipeline and the robotic controller, while Google Sheets operates as a queue-based database storing commands, parameters and execution status. This allows asynchronous data exchange between the design environment and the physical system.



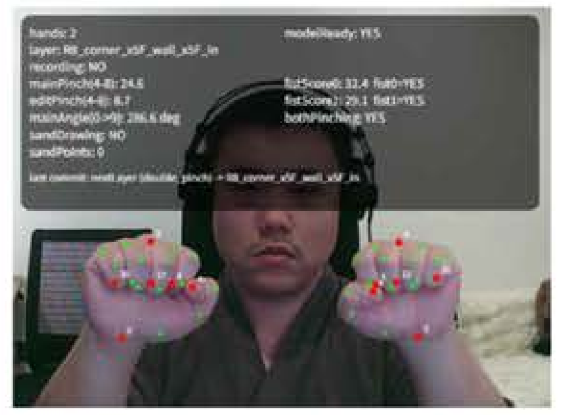
Database



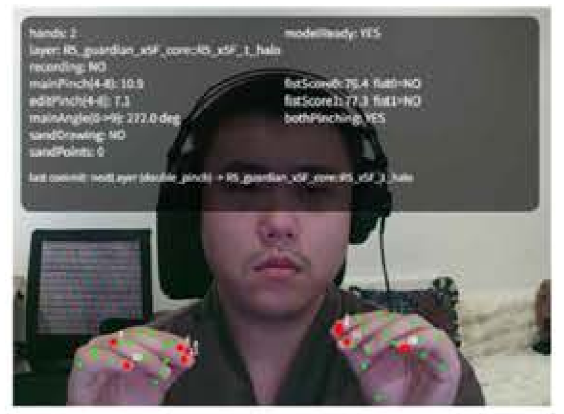
Gesture-Based Interaction

Several gesture mappings are implemented to enable intuitive control of the sand-deposition system. Each gesture is mapped to a specific operational command within the robotic workflow.

***Both hands clenched (fists)* → *execute Draw All*, triggering the full drawing sequence.**



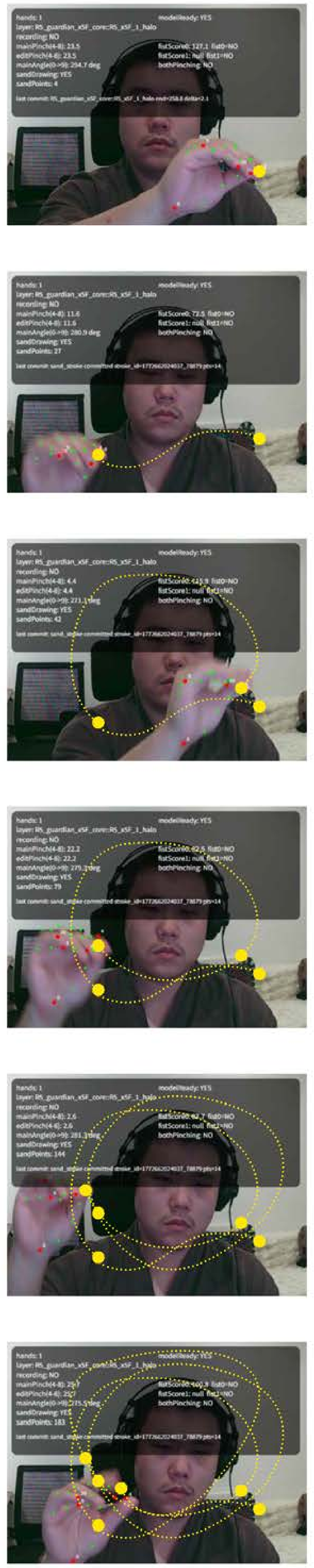
***Both hands pinching* → *switch layers*, allowing the user to move between different drawing layers.**

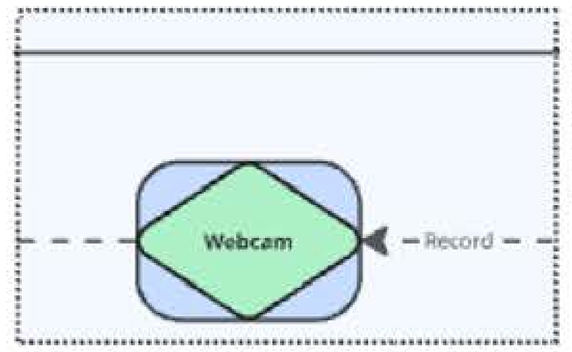


***Double-hand rotation around an anchor point* → *edit stroke paths*, enabling adjustments to existing trajectories.**

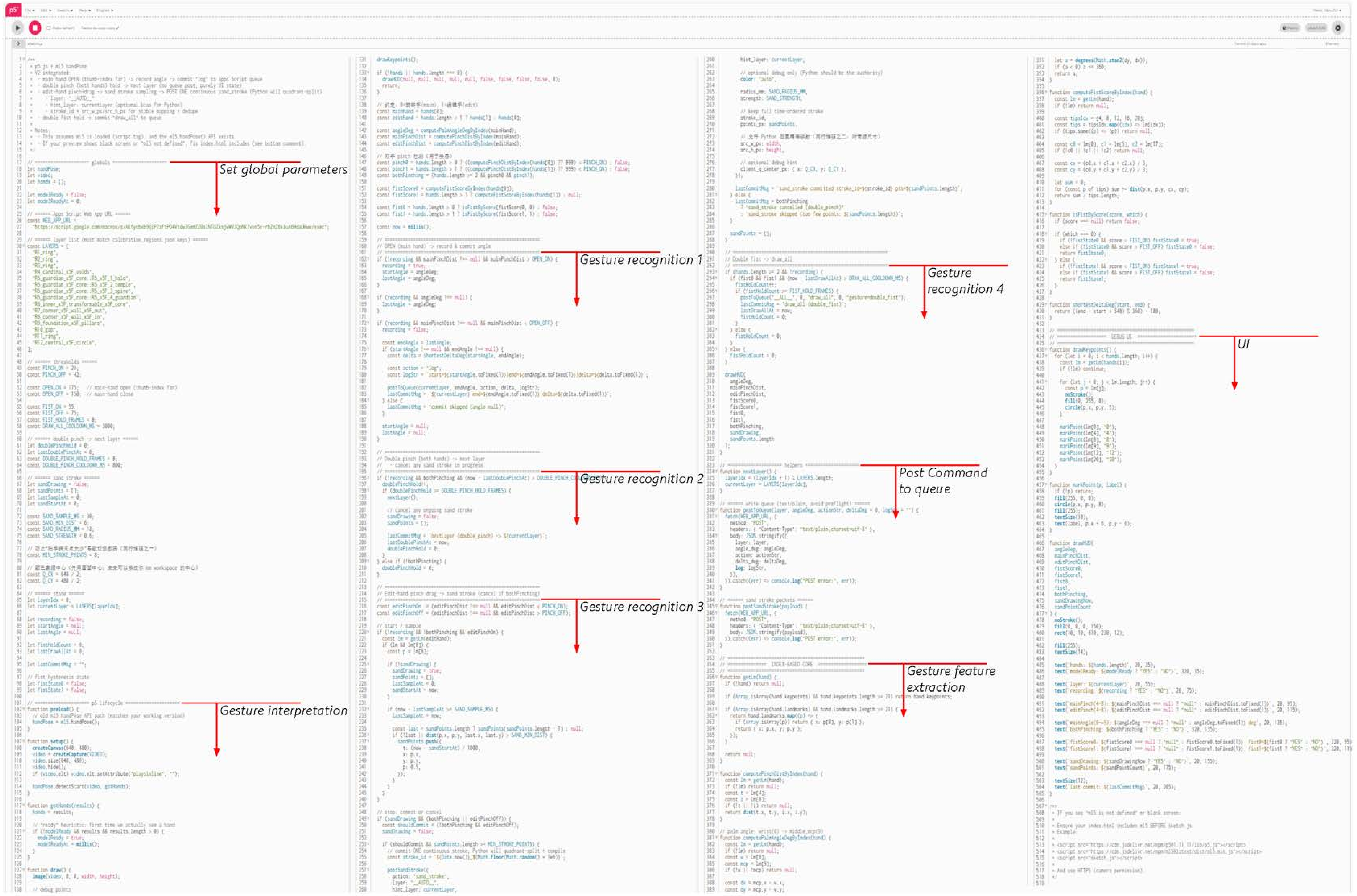
***Single-hand rotation around an anchor point* → *edit stroke paths*, enabling adjustments to existing trajectories.**

***Single-hand pinch and drag* → *sand deposition*, controlling the placement and movement of sand in real time.**

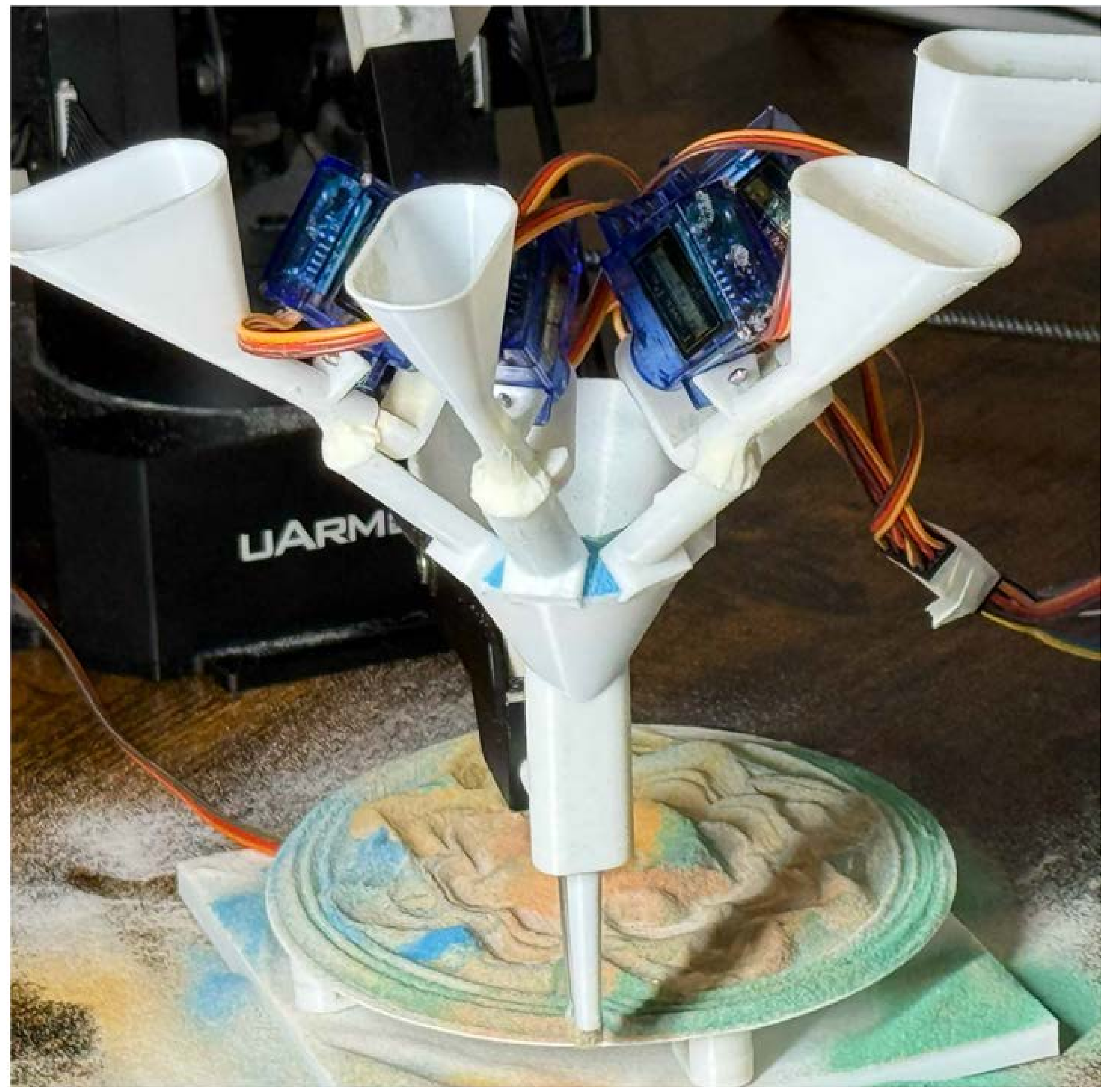




User Interaction

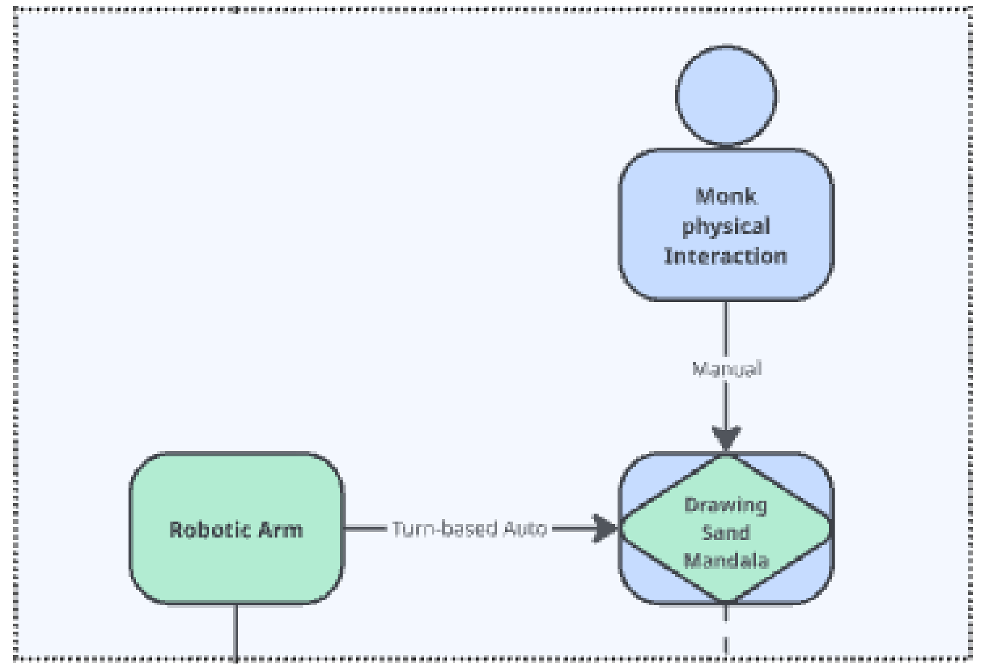


A browser-based p5.js interface captures hand landmarks from a webcam and analyses their relative positions, orientation and movement. These features are classified into gesture commands, packaged as structured JSON data and transmitted through HTTP requests to the cloud queue for subsequent robotic execution. The interface therefore acts as the human–machine interaction layer between bodily input and machine-readable control.



This stage integrates the full control system with the robotic sand-deposition apparatus. Commands are read from the cloud queue, processed through Python, and translated through an Arduino-based control layer into robotic motion, valve actuation and colour selection. The resulting workflow converts structured digital instructions into physical sand deposition.

Scan / click *to view test video*



Physical Application

Set global parameters

System initialization

Serial command listener

Close all valves

Open sertain valves

ESP32 Dev Module

sketch_feb4a_aRDUINO.ino

```
#include <Wire.h>
#include <Adafruit_PWMServoDriver.h>

Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver(0x40);

#define SERVO_MIN_TICK  150
#define SERVO_MAX_TICK  520

#define CH_RED      12
#define CH_GREEN    13
#define CH_BLUE     14
#define CH_YELLOW   15

#define OPEN_US     1600
#define CLOSE_US    2100

#define I2C_SDA     21
#define I2C_SCL     22

static inline uint16_t clampU16(int v, int lo, int hi) {
  if (v < lo) return lo;
  if (v > hi) return hi;
  return (uint16_t)v;
}

uint16_t usToTicks(int us) {
  long t = map(us, 500, 2500, SERVO_MIN_TICK, SERVO_MAX_TICK);
  return clampU16((int)t, SERVO_MIN_TICK, SERVO_MAX_TICK);
}

void servoOpen(int ch)  { pwm.setPWM(ch, 0, usToTicks(OPEN_US)); }
void servoClose(int ch) { pwm.setPWM(ch, 0, usToTicks(CLOSE_US)); }

void closeAll() {
  servoClose(CH_RED);
  servoClose(CH_GREEN);
  servoClose(CH_BLUE);
  servoClose(CH_YELLOW);
}

String line;

void printHelp() {
  Serial.println();
  Serial.println("PCA9685 Valve Controller Ready.");
  Serial.println("Commands:");
  Serial.println("  open_only red|green|blue|yellow");
  Serial.println("  close_all");
  Serial.println("  help");
  Serial.println();
}

void setup() {
  Serial.begin(115200);
  delay(300);

  Wire.begin(I2C_SDA, I2C_SCL);

  pwm.begin();
  pwm.setPWMFreq(50);
  delay(10);

  closeAll();
  printHelp();
}

void loop() {
  while (Serial.available()) {
    char c = (char)Serial.read();
    if (c == '\r') continue;

    if (c == '\n') {
      line.trim();
      if (line.length() == 0) { line = ""; continue; }

      if (line == "help") {
        printHelp();
      }
      else if (line == "close_all") {
        closeAll();
        Serial.println("ALL CLOSED");
      }
      else if (line.startsWith("open_only")) {
        closeAll();

        if (line.indexOf("red") >= 0)        servoOpen(CH_RED);
        else if (line.indexOf("green") >= 0) servoOpen(CH_GREEN);
        else if (line.indexOf("blue") >= 0)  servoOpen(CH_BLUE);
        else if (line.indexOf("yellow") >= 0)servoOpen(CH_YELLOW);
        else {
          Serial.println("ERR: unknown color");
          line = "";
          continue;
        }

        Serial.println(line);
      }
      else {
        Serial.println("ERR: unknown command (type: help)");
      }

      line = "";
    } else {
      line += c;
      if (line.length() > 80) line.remove(80);
    }
  }
}
```

PC-side runtime configuration

Valve upstream control

Robotic Arm control

Geometry processing + robot execution

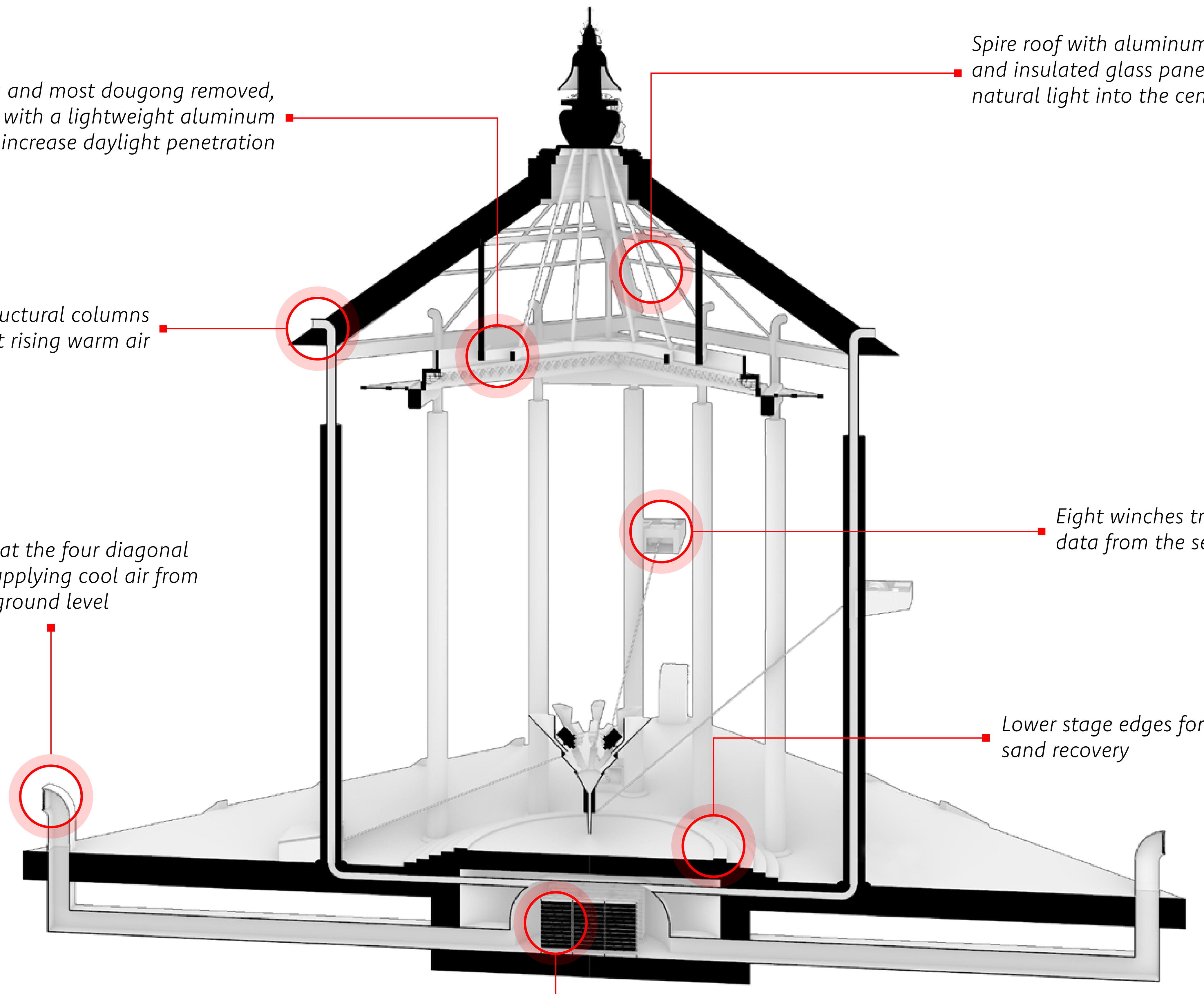
Sand stroke compiler + vulve execution

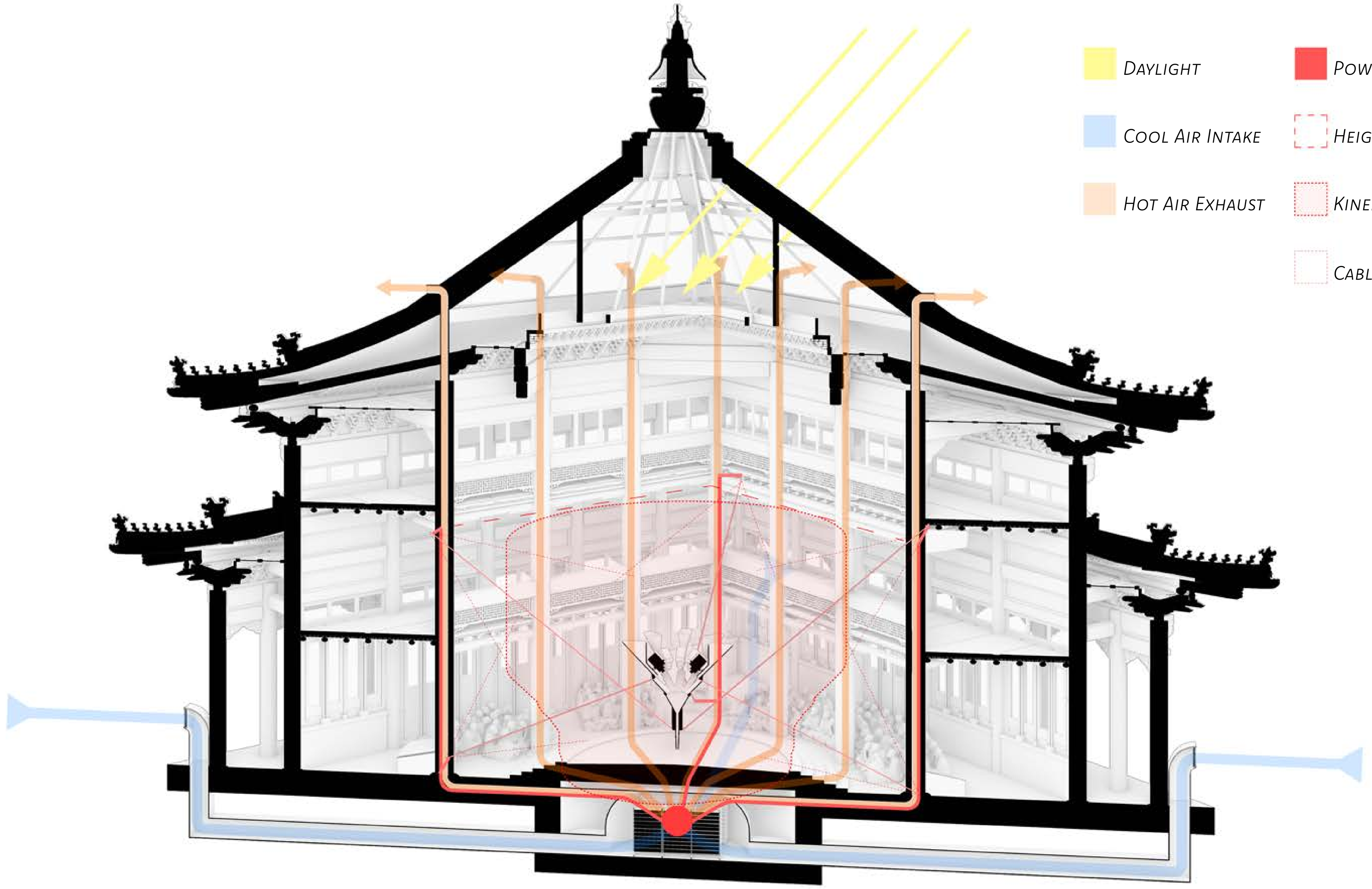
Read guestures from spreadsheet

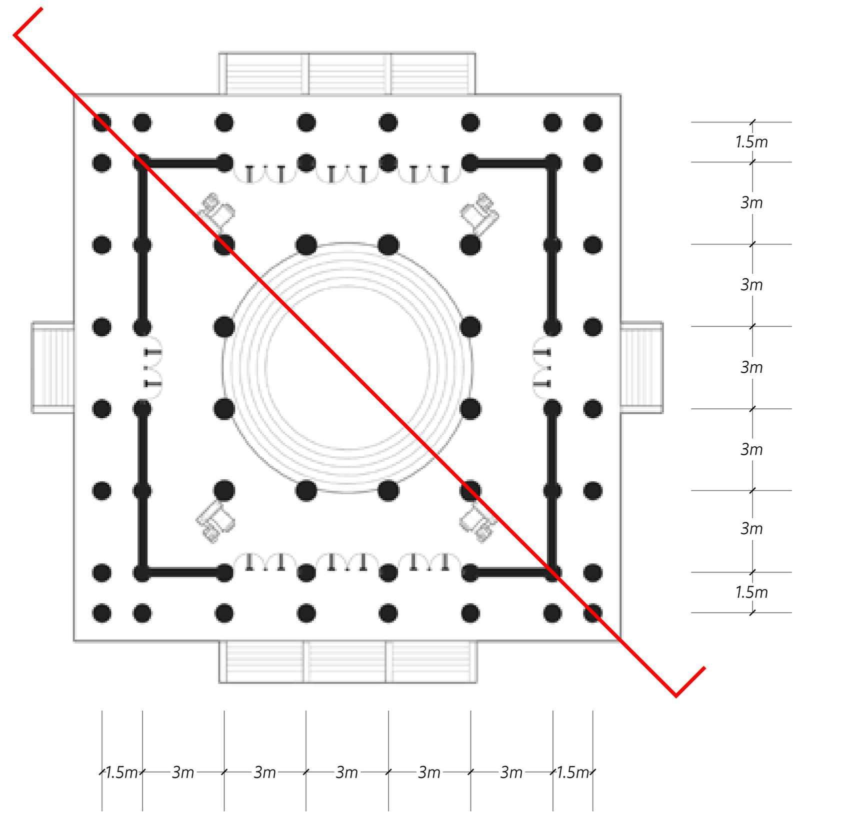
Loop of data pulling and command sending

1.3 Architectural System Integration

The robotic mandala system is integrated through a series of targeted architectural modifications. Roof alterations improve daylight penetration, hollow structural columns support passive heat exhaust, and dedicated air, power and data routes connect the server room with the robotic apparatus.







Together, these interventions coordinate robotic movement, environmental control and technical infrastructure within a single architectural framework, allowing the computational system to operate as part of the building rather than as a separate technical layer.

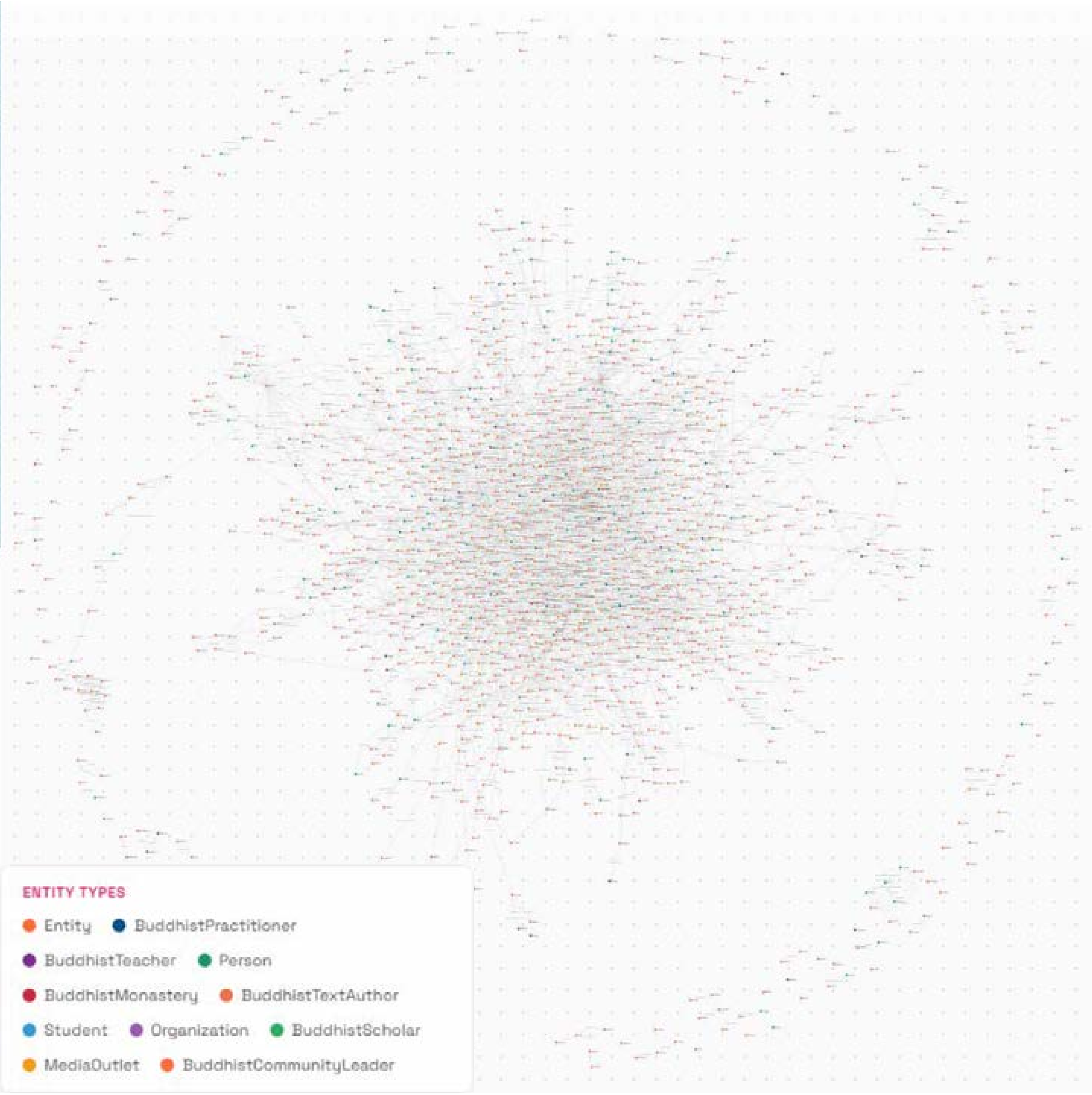
1.4 AI-Driven Responsive Media

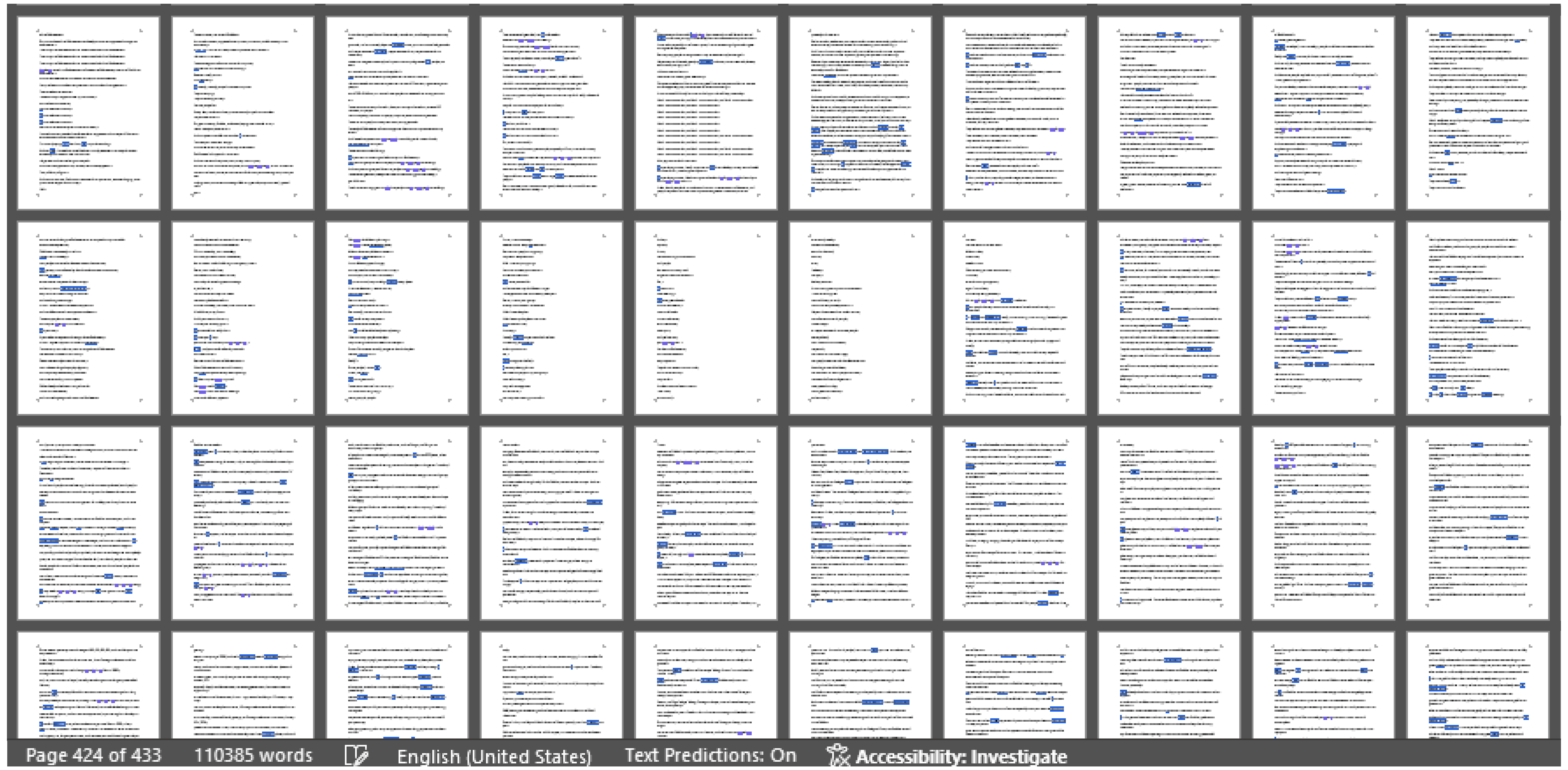
Knowledge Archive & Data Preparation
Knowledge Analysis & AI Agent
Visual Library & Physical Implementation
Integrated Spatial Proposal

How can knowledge become an architectural material?

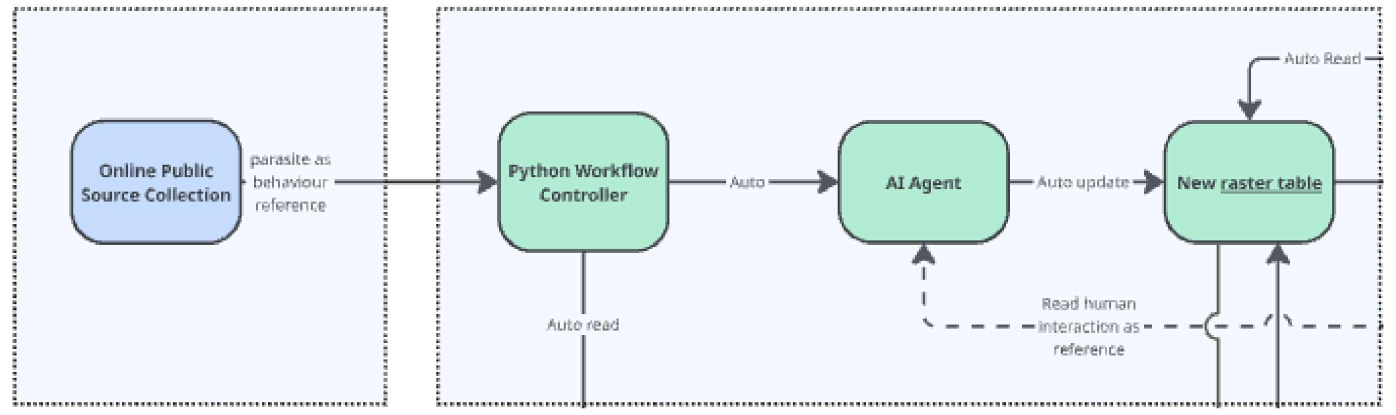
The project tests a framework in which cultural knowledge is archived as structured data, interpreted by AI, and returned to space through generative media. Using Tibetan Buddhist teachings as a case study, a machine-readable knowledge base is connected to a locally deployed AI system and Seedance pipeline, producing a continuously evolving, content-responsive mural.

The workflow was implemented as a custom Python-based agent system, coordinating document processing, knowledge retrieval, LLM interpretation and downstream media generation within a single pipeline.



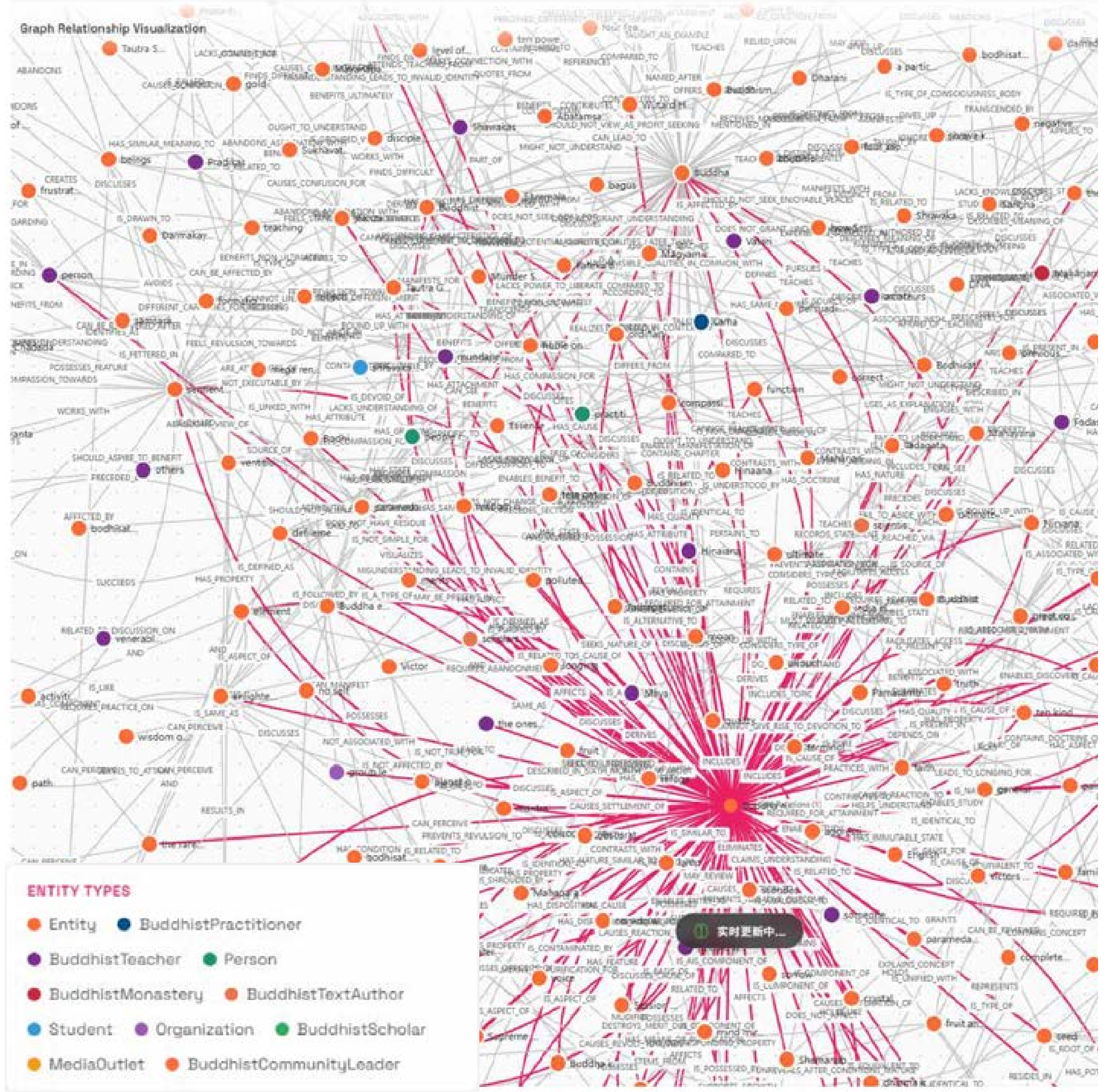


A corpus of public lectures by Khenpo Sodargye, one of the senior teaching figures at the site, was collected from online platforms, transcribed and translated into a machine-readable textual archive. The resulting dataset exceeds 430 pages and 110,000 words, forming the primary knowledge source for the subsequent AI workflow.



Text Extraction -> File Analysis & Structuring -> Python-Based Orchestration -> LLM Reasoning

The structured lecture corpus is analysed through Zep to identify and organise recurring concepts, entities and relationships. These outputs are then incorporated into a custom Python-based AI agent, which retrieves relevant contextual material and coordinates LLM responses within the wider generative workflow.



Python-Based AI Agent

retrieve relevant material
assemble context
send to LLM
generate contextual response
return structured output

Zep Based Analysis

corpus / document ingestion
concept extraction
entity / relationship analysis
knowledge structure

```
reasoner.py ×
reasoner.py > ...
def format_evidence_block(ctx: dict) -> str:
            src = e.get("source_name", "") or e.get("source", "") or "?"
            rel = e.get("name", "") or "RELATES_TO"
            tgt = e.get("target_name", "") or e.get("target", "") or "?"
            fact = (e.get("fact") or "").strip().replace("\n", " ")
            line = f"- {src} --{rel}--> {tgt}"
            if fact:
                line += f" | fact: {fact[:180]}"
            parts.append(line)

    if matched_episodes:
        parts.append("\nEvidence passages:")
        for i, ep in enumerate(matched_episodes, 1):
            text = (
                ep.get("text")
                or ep.get("content")
                or ep.get("summary")
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS
PS C:\Users\98371\Desktop\AI_Buddhism> python run_buddhist_reasoner.py

=== ANSWER ===

One-sentence essence:
The pursuit of wealth is a human preoccupation that can be understood through the lens of

Direct answer:
The path to wealth is often fraught with the desire for more, which can lead to attachmen
asting values.

Buddhist framing:
In Buddhism, the desire for wealth can be seen as a manifestation of craving, one of the
Cultivating compassion and understanding the interdependent nature of all phenomena can h
```

Custom Python-based AI agent and media pipeline

- display_player — *Manages live playback and updates*
  - __pycache__
  - __init__.py
  - play_history_then_live.py
  - watch_and_postprocess.py
  - watch_display_loop.py
- input_gateway — *Captures and transcribes input*
  - __pycache__
  - audio
  - clients
  - input_logs
  - server.py
  - settings.py
  - storage.py
  - transcribe_whisper.py
- pipeline — *coordinates system logic and data flow*
  - __pycache__
  - outputs
  - run_live_pipeline.py
  - test_latest_pipeline.py
- reasoner — *retrieves context and selects visual states*
  - __pycache__
  - image_selector.py
  - packet_reader.py
  - test_directed_selection.py
  - test_latest_packet.py
  - test_select_image.py
  - visual_state_reasoner.py
- runtime_logs — *Monitors system execution*
  - display_loop.log
  - input_gateway.log
  - live_pipeline.log
  - s3_ambient.log
  - seedance_watcher.log
  - watch_and_postprocess.log
- video_generation — *Sends prompts to Seedance and post-processes outputs*
  - __pycache__
  - keyframes
  - seedance_outputs
  - state
  - .env
  - 000_start.png
  - postprocess_latest_video.py
  - seedance_from_pipeline.py
  - seedance_interactive.py
  - watch_seedance_from_pipeline.py
- visual_library — *Organises indexed visual references and mural states*
  - images
  - mural_index.json
- .txt
- system_launcher_final.py — *Runtime coordinater*

The final system was implemented as a modular Python application rather than a single generative script, coordinating input capture, contextual reasoning, visual selection, Seedance generation, post-processing and live playback across a unified runtime.

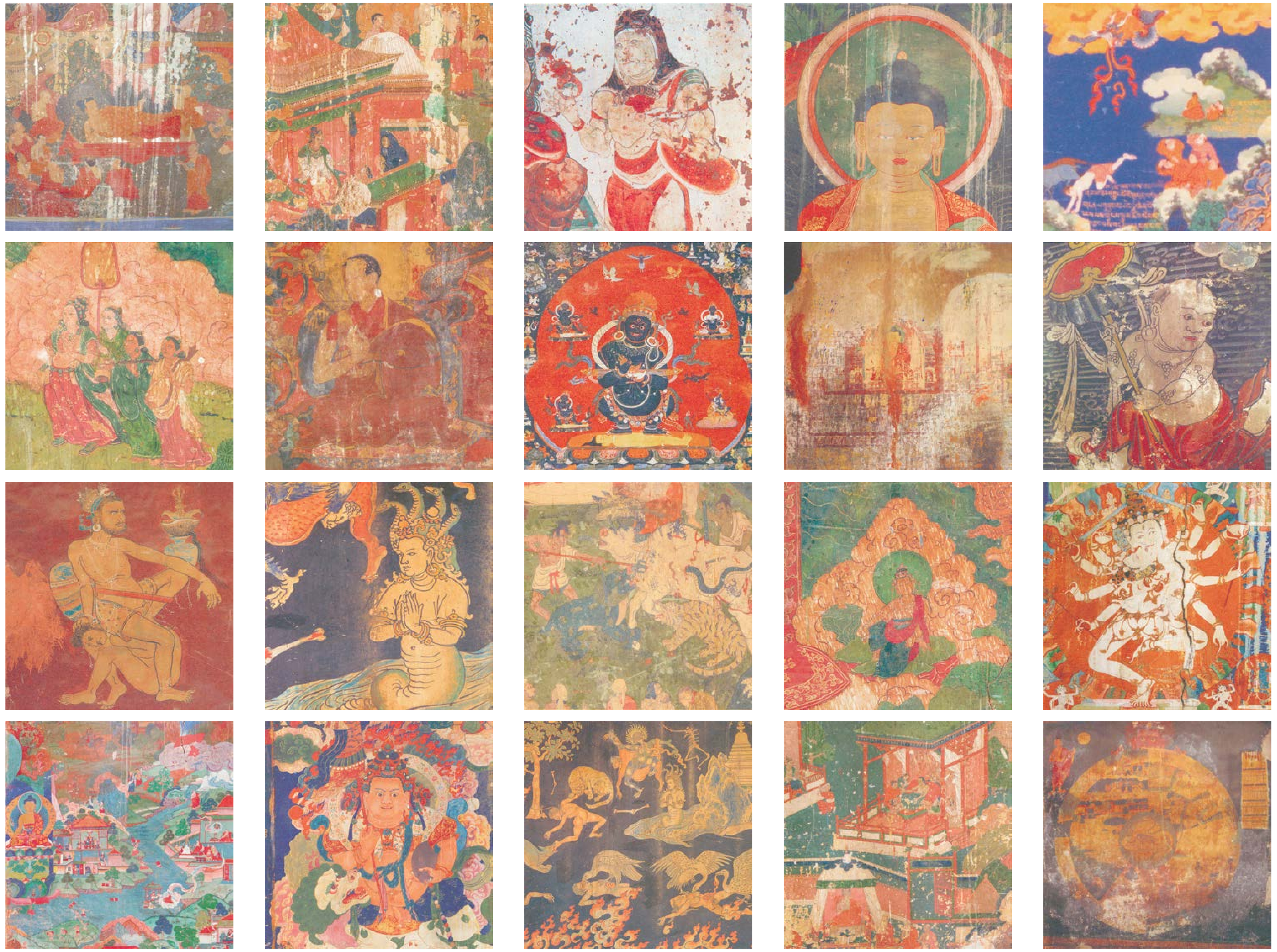
Visual Library

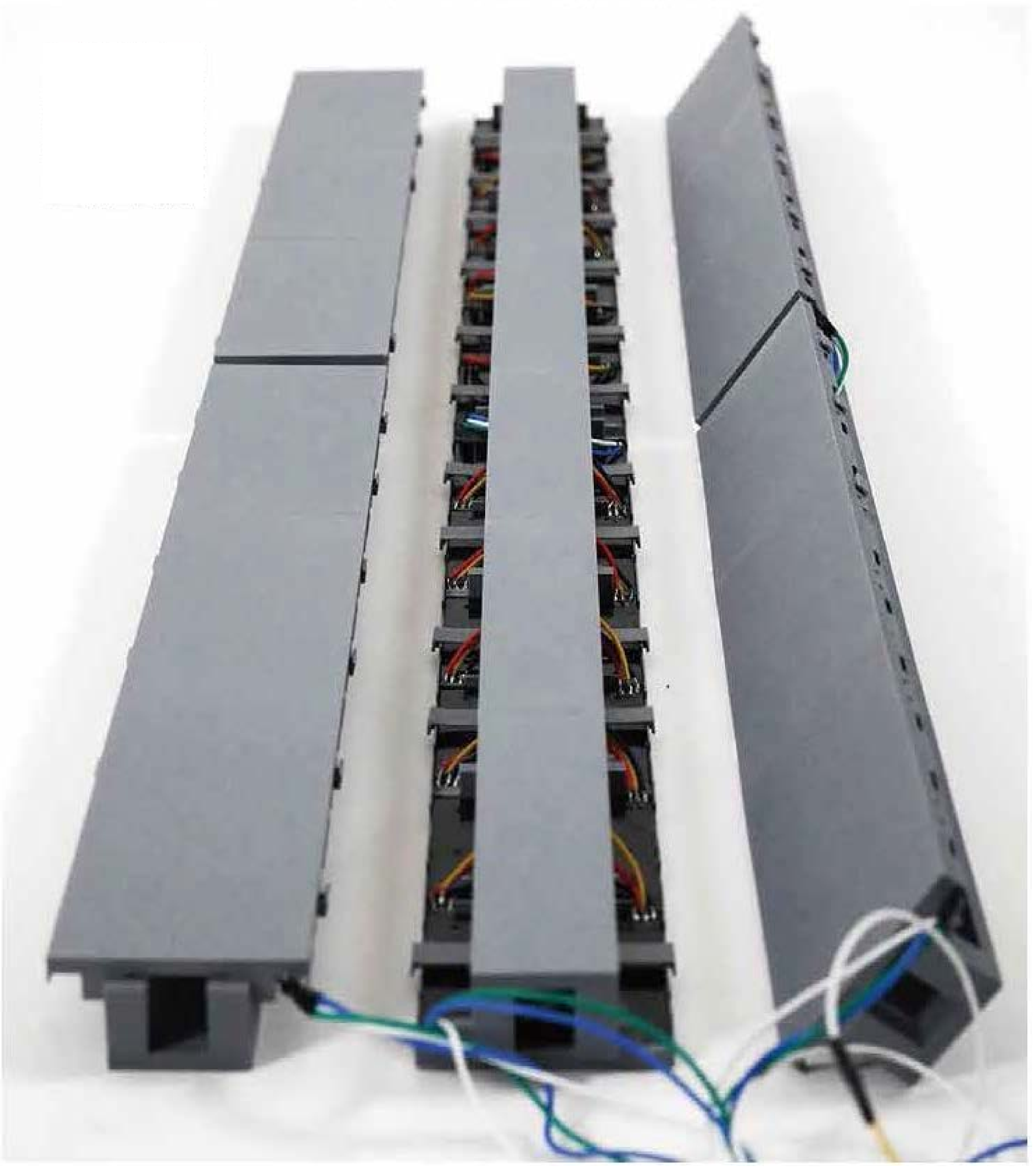
The visual library draws on mural imagery documented at Gongga Qude Monastery in Tibet and is organised through a Buddhist conceptual framework. These indexed references provide the visual basis for the generative media system.

Scan / click
to view the dynamic mural archive developed through iterative physical testing

While the Robotic Sand Mandala translates a cultural practice into machine-readable geometry and physical fabrication, the AI-Driven
Responsive Media operates on cultural knowledge itself. Textual teachings are archived, interpreted through AI and transformed into evolving visual content, allowing architecture to act not only as a container for knowledge, but as a medium through which that knowledge can be continuously re-expressed.

Arising | *Abiding* | *Dissolution* | *Emptiness* | *Transformation*





1:1 Architectural Media Prototype

The prototype demonstrates how dynamically generated visual content can operate as an architectural media surface rather than as a standalone screen-based output.

Scan / click
Composed Film
(3 mins)

Robotic Sand Mandala

During the annual sand mandala ritual, the system enables a mandala master at Gonggar to transmit drawing instructions to the robotic installation in London. Students at the remote site can participate in the same making process rather than observing a digital reproduction. The system therefore preserves the mandala as an embodied and repeatable practice, while extending participation across geographical distance.

AI-Driven Responsive Mural

During daily chanting at Gonggar, spoken teachings and contextual inputs are processed through the AI system and translated into evolving mural imagery in London. Rather than reproducing a fixed image or recording, the architectural surface changes in response to the knowledge being transmitted. The system therefore extends the communication of Buddhist teaching through a dynamic spatial medium while retaining a connection to its living source.

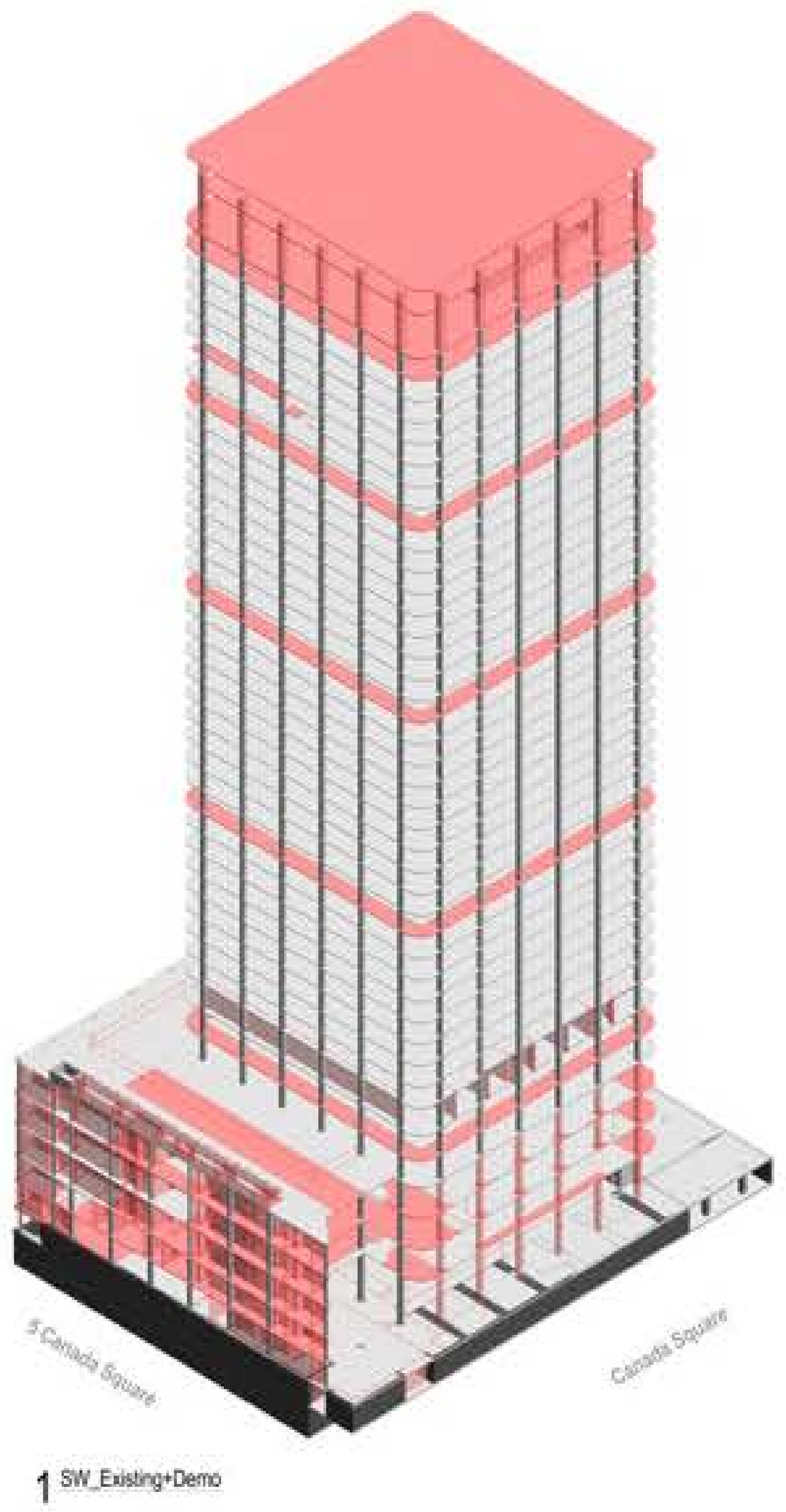
2. Professional Practice

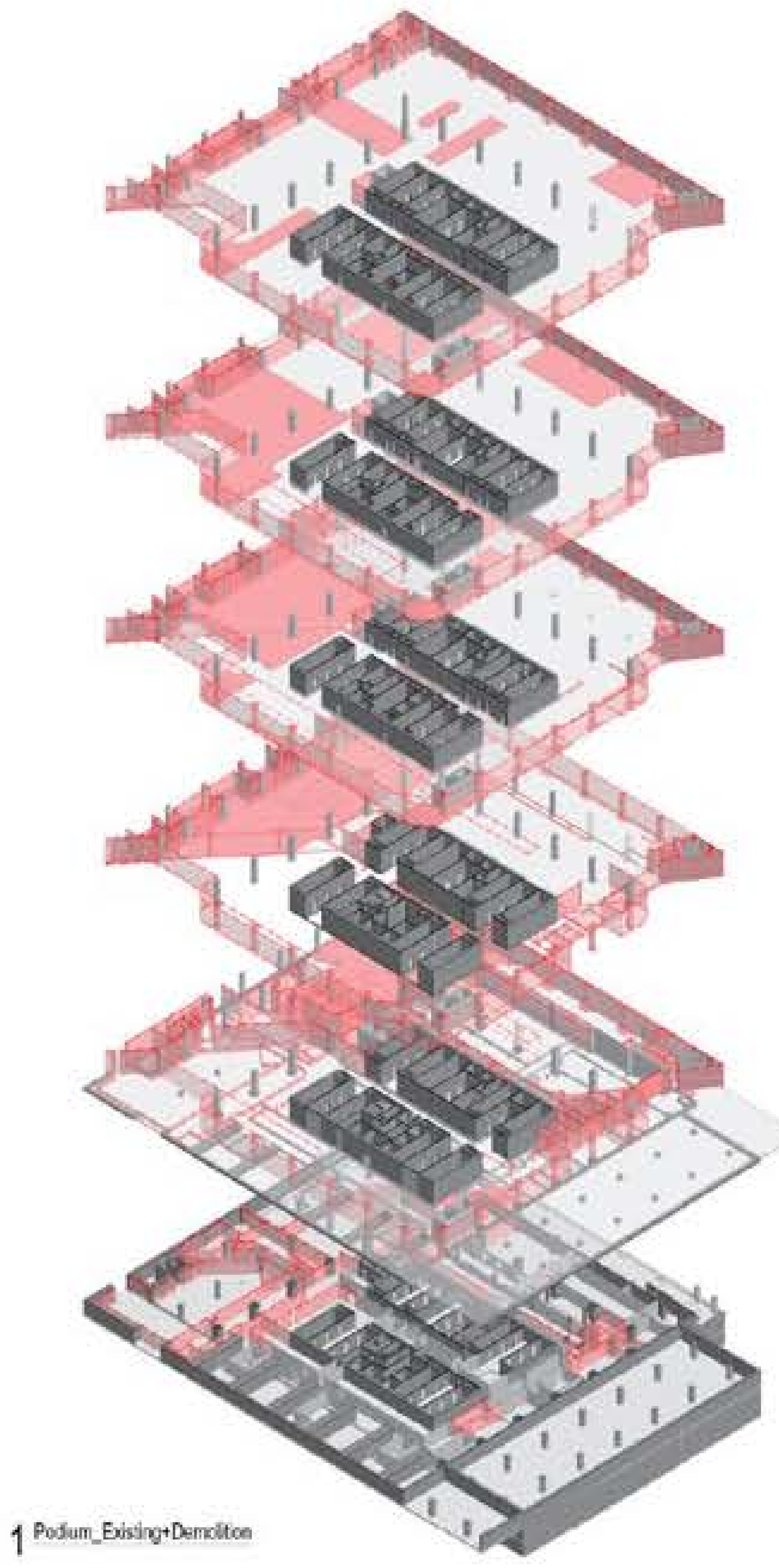
Working experience at Foster+Partners, Studio 1

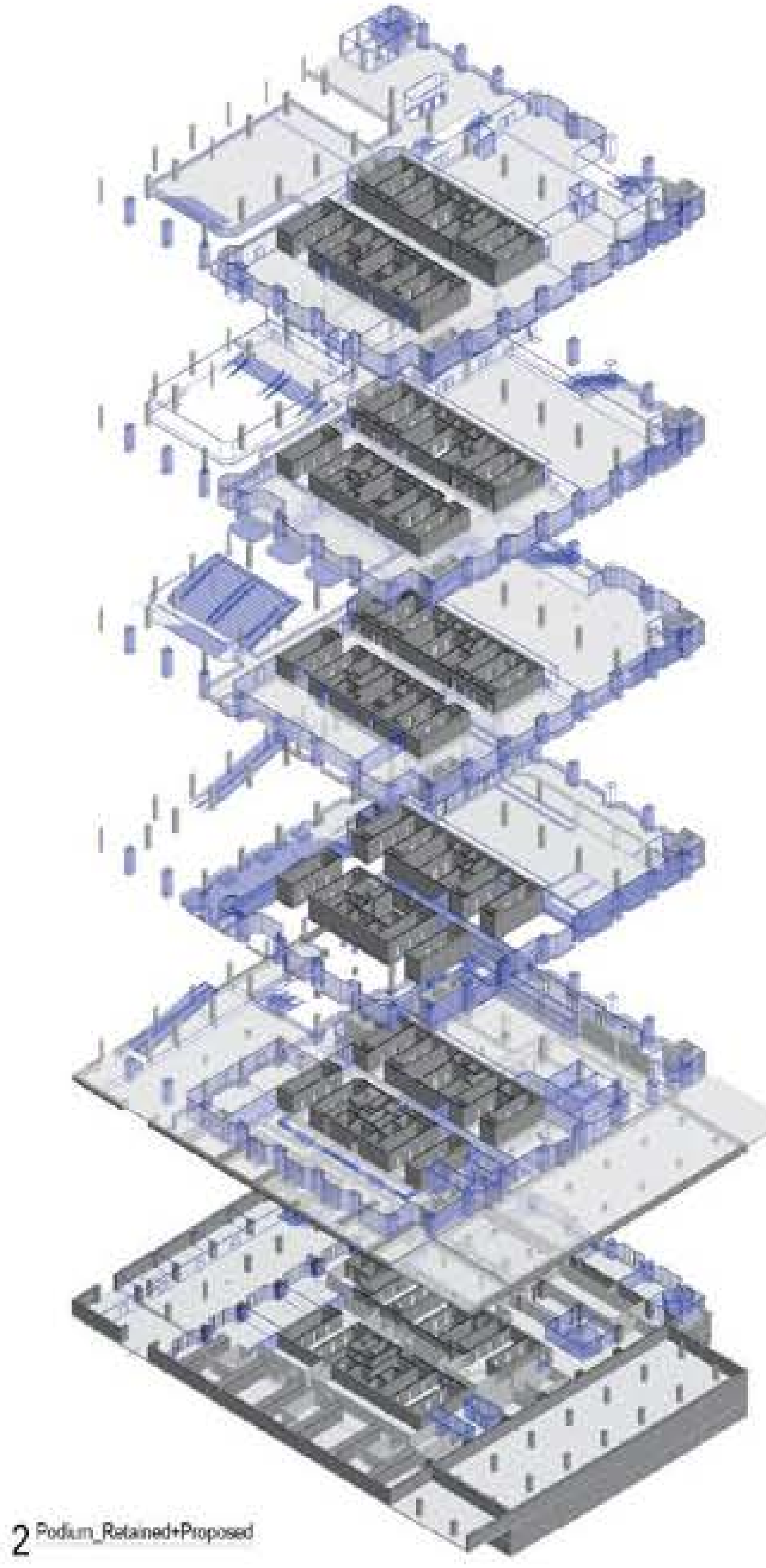
2.1 – 8 Canada Square

(Rhino, Unreal Engine, Revit)

Developed as a competition proposal, the project involved the transformation of the existing tower and its upper-level public spaces. My contribution included taking primary responsibility for the Revit model and developing the cinematic visualisation pipeline for the disc-shaped exhibition space at the top of the tower, alongside comparative modelling of retained, demolished and proposed elements.







Podium – Exhibition Room

Tower Top – Observation Deck I

Tower Top – Observation Deck II

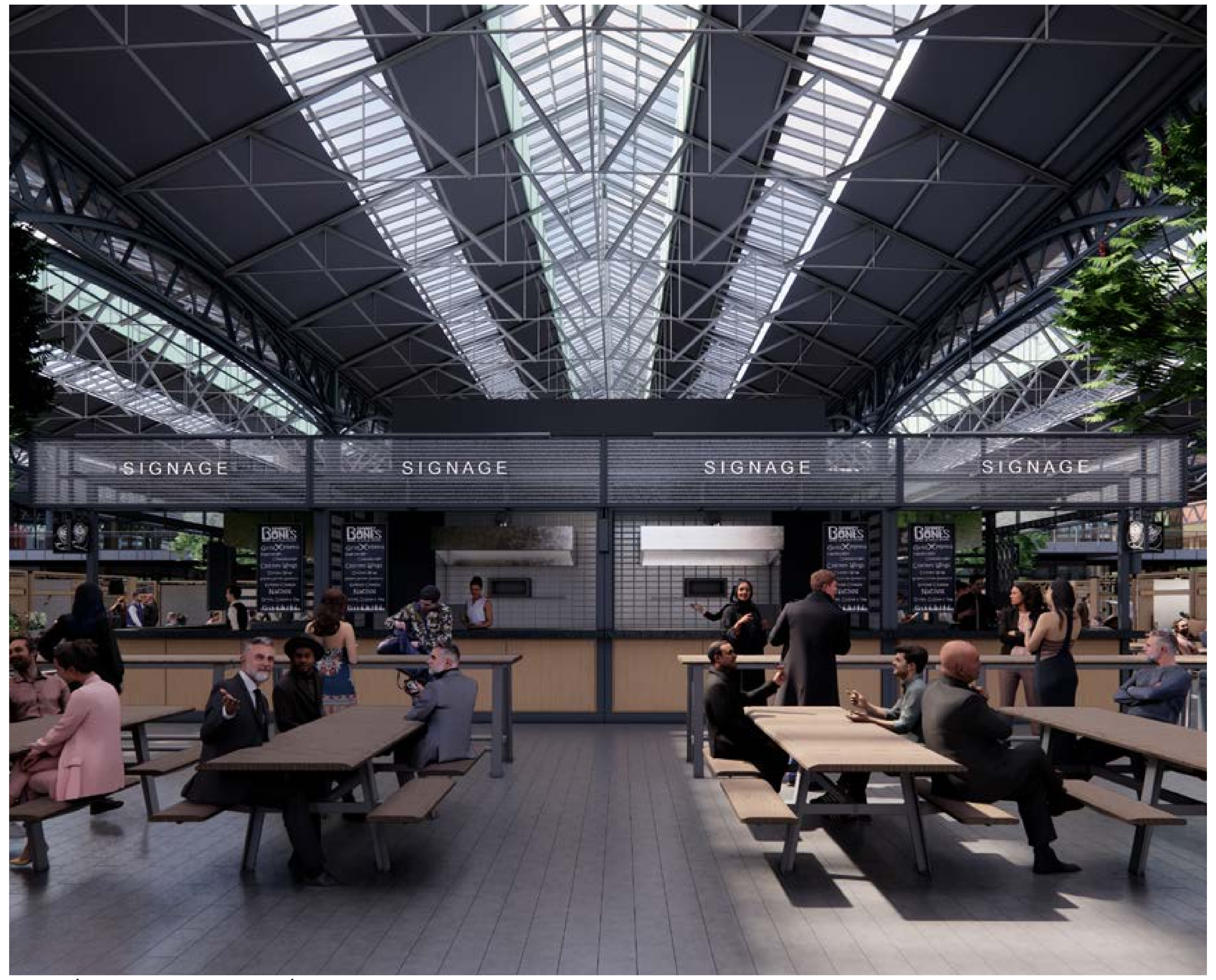
2.2 – 1 Spitalfields

(Rhino, Enscape, Revit, CAD)

Main Building – Proposed Perspective

I worked on 1 Spitalfields through its transition from RIBA Stage 2 to Stage 3. My contribution included market and affordable workspace design, early below-ground studies, project visualisation, and extensive Revit drawing production and coordination across general arrangement and detailed packages.

Market – Proposed View I



Market – Proposed View II

Main Building – Proposed Revit Drawings

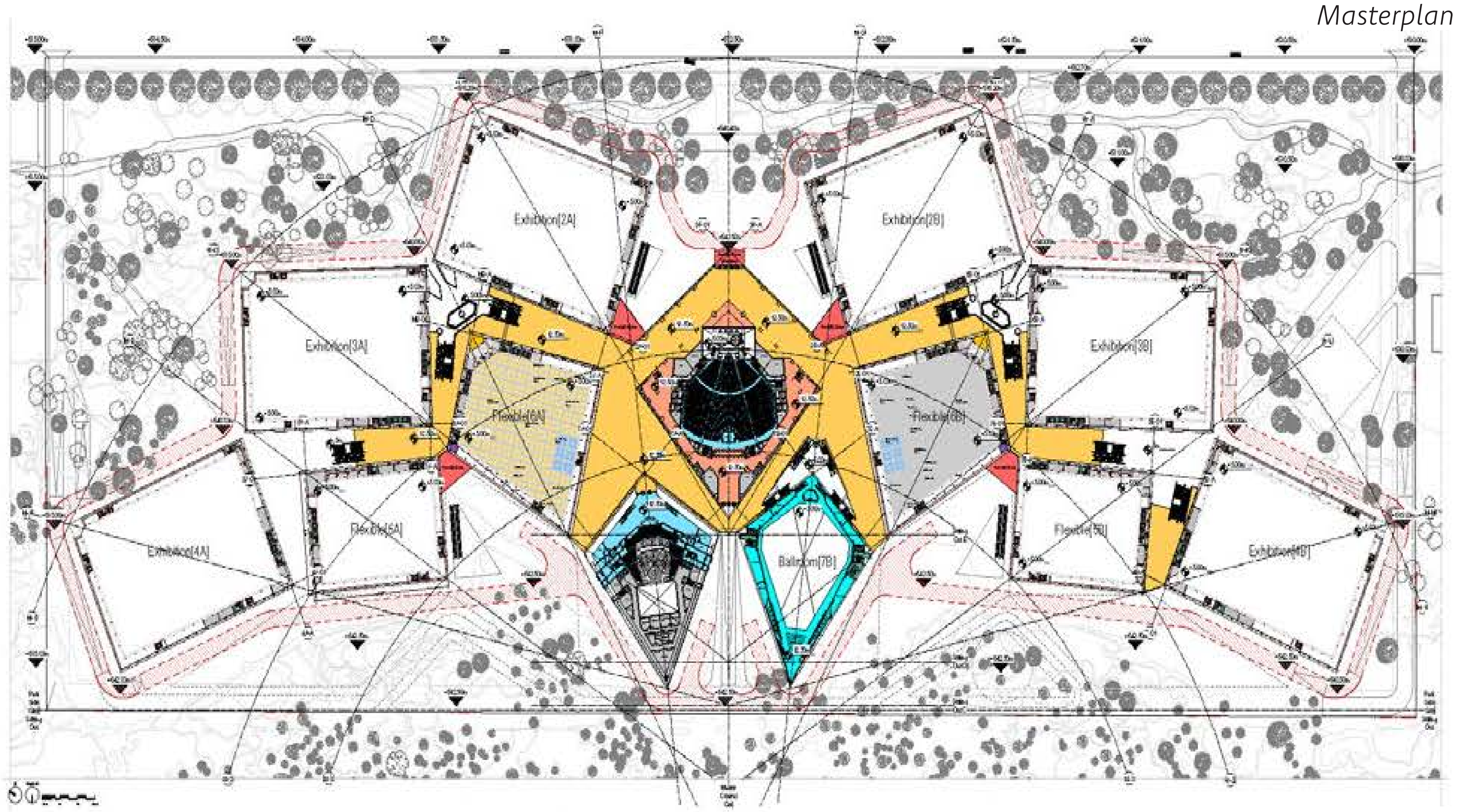
2.3 - RECC

(Rhino, Grasshopper, Enscape, Revit, CAD)

Developed at RIBA Stage 2 within a team of more than twenty, the project involved frequent design iterations, coordination and visualisation studies. My contribution included Revit development, design coordination and internal reviews, with particular involvement in the mezzanine and sunken communal areas.

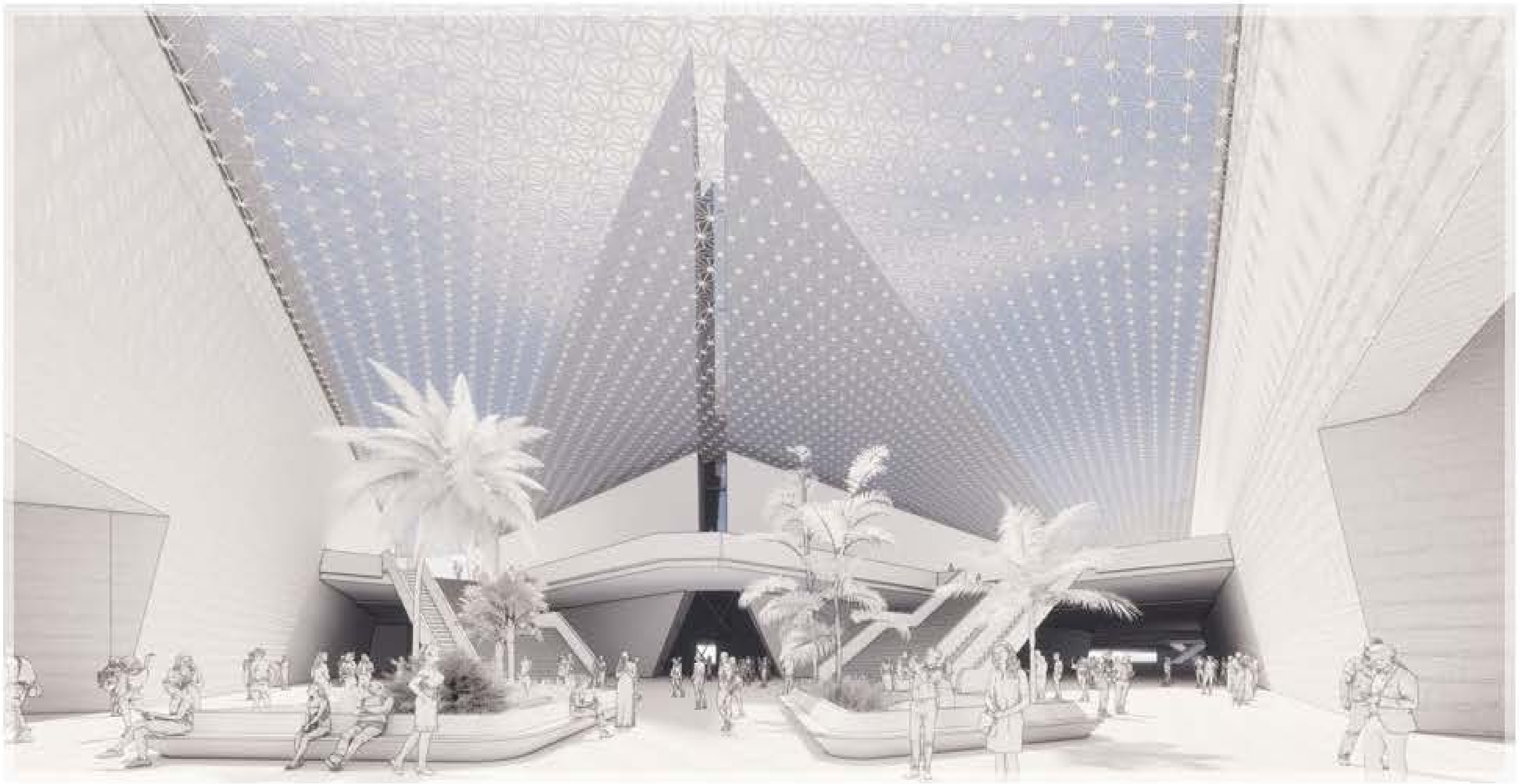
I also contributed to the build-up and setup of a Grasshopper-based parametric shading system, helping structure the geometric logic and test design variations within the façade strategy.

Masterplan



Mezzanine Area

Sunken Communal Area

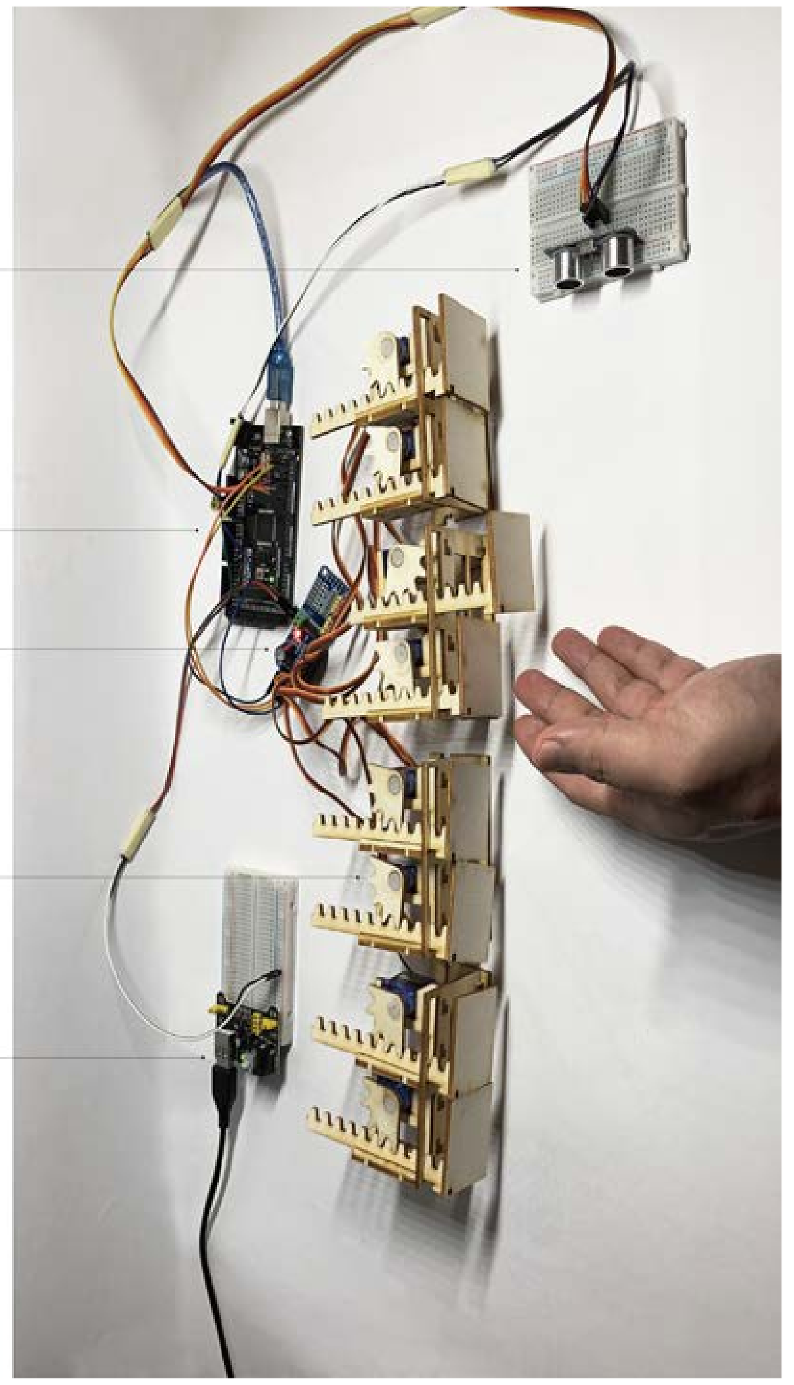


3. Other Works

Selected Computational Studies

3.1 Light Up New York For But A Moment

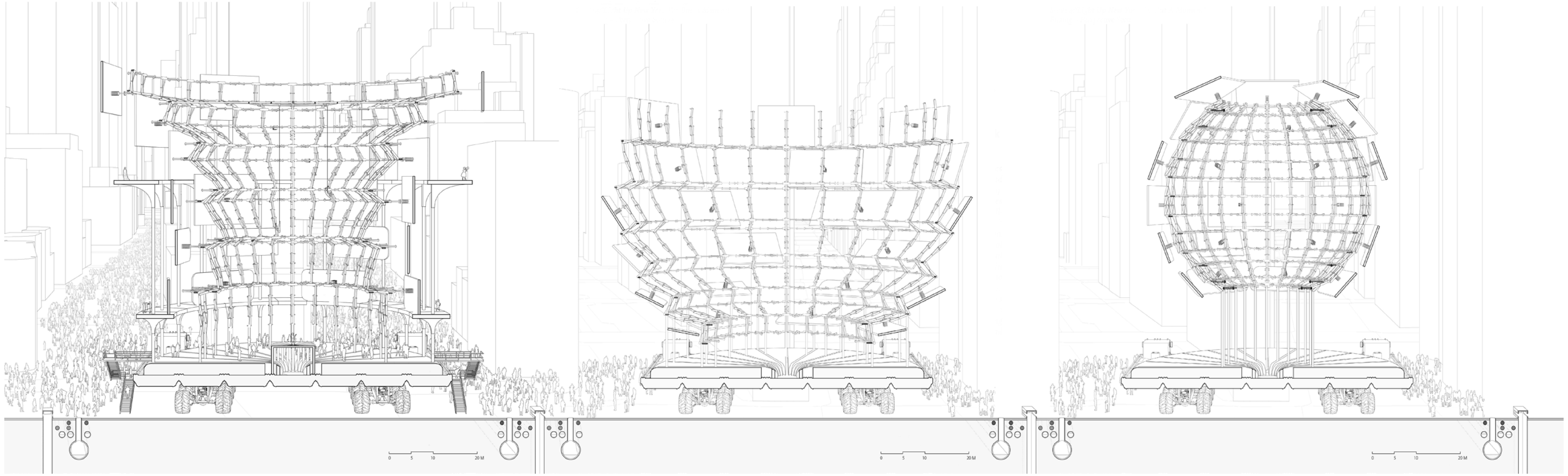
Rhino, Unreal Engine, Adobe Creative Cloud, Arduino



The project develops a large-scale transformable structure for Times Square, using a consistent primary frame to support multiple spatial configurations over time. Through sectional studies, iterative structural modelling and real-time visualisation, the system was tested across open, enclosed and illuminated states. Advertising screens, circulation and structure are integrated into a single inhabitable public framework rather than treated as separate attachments.

A physical control prototype was also developed to test interactive transformation logic between digital input and spatial response.

Final film (8 mins)

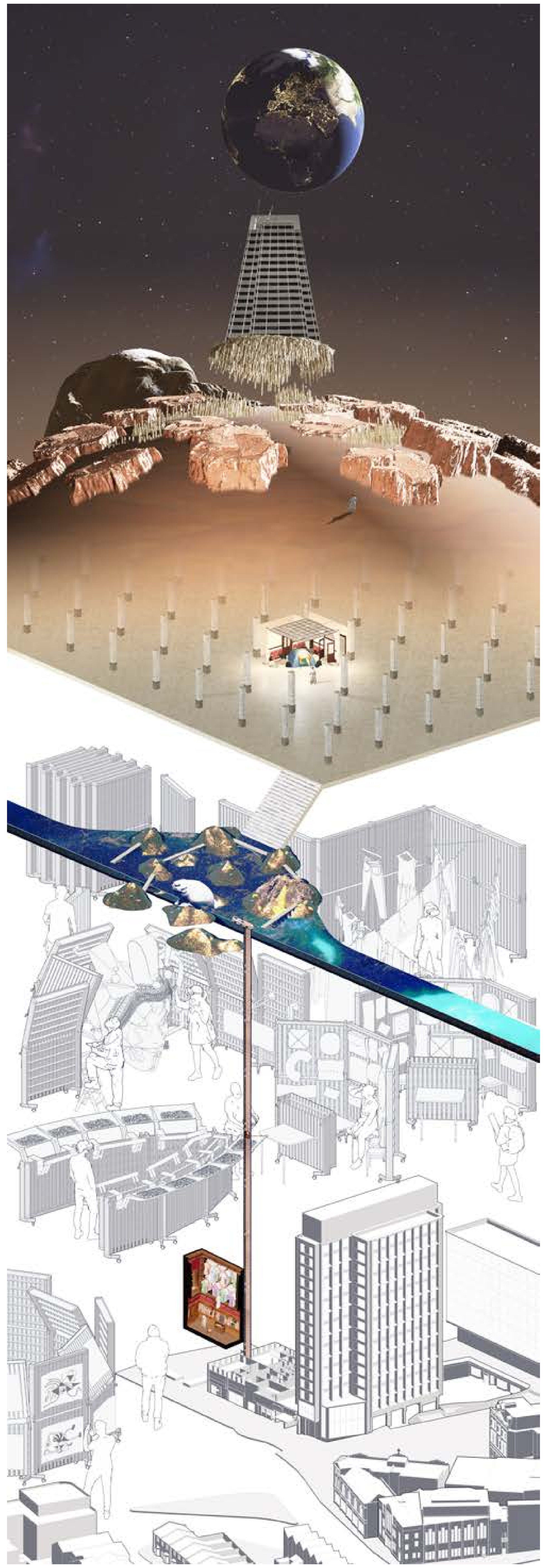


3.2 No Curtain Call

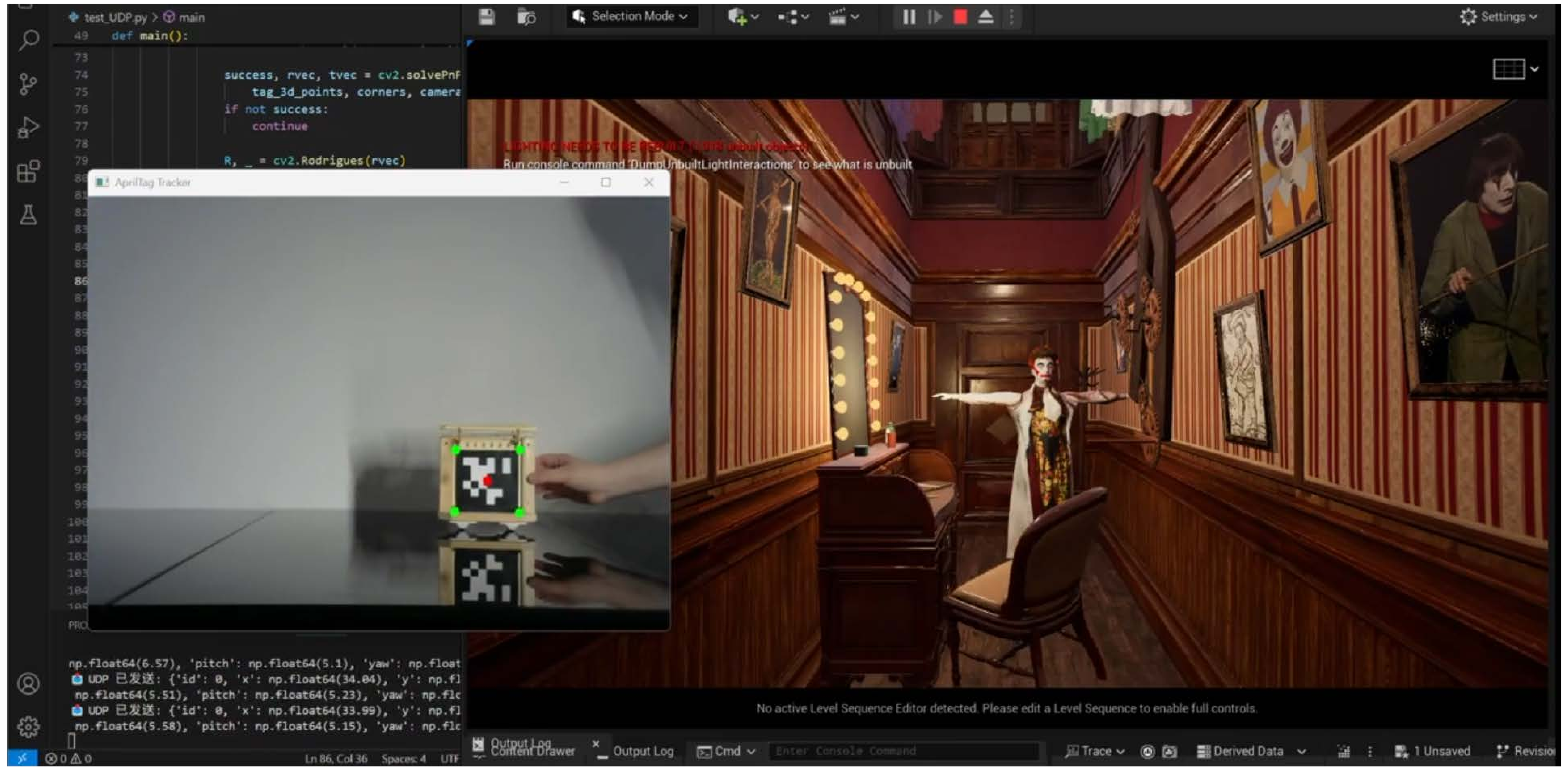
VR Interactive Theatre

No Curtain Call, developed in my fourth year, explored a real-time performance workflow connecting inertial motion capture with Unreal Engine. I became familiar with live character tracking, motion-data transfer and real-time visualisation, while also developing a custom AprilTag-based interaction system through Unreal Engine Blueprints. The system used image recognition to track physical objects and map their movement into the digital environment.

For example, moving a chair in the physical space simultaneously repositioned its digital twin in Unreal Engine. This created a direct interaction layer between physical performance and the virtual scene.



Testing Video



AprilTag Object Tracking -> Digital Twin Interaction

VR Performance

Inertial Motion Capture -> Real-Time Performance