



PYTHON AND SEMANTIC TECHNOLOGIES

Applying Semantic Technologies to Widely Diverse Endeavours
– Brett Alistair Kromkamp for **PyCon Sweden 2020**

WHO AM I?

BRIEF OVERVIEW

- **Brett** Alistair Kromkamp – @brettkromkamp (Twitter)
- Dutch, born in Africa (Zambia), living in Northern **Norway**
- Primarily a **software developer**, but also a CTO for 4 years and currently, CMO
- **Semantic technologies** solutions provider
- Worked in the **tourism industry** in Spain as a software developer and Team Lead for 17 years
- Currently, working in the **EdTech** sector – and have been, for the last 8 years



**THE TOPIC MAPS
PARADIGM**

01

**USE CASE 1: KNOWLEDGE
MANAGEMENT**

02

**USE CASE 2:
STORYTELLING**

03

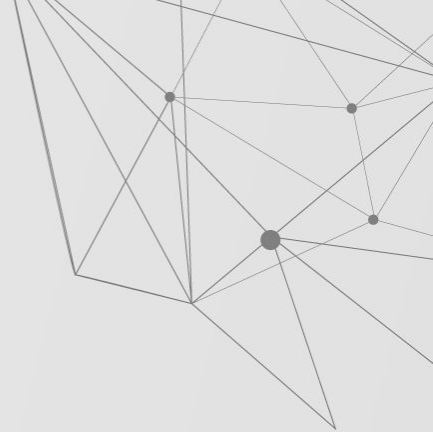
TALKING POINTS

04

**THE STORYTELLER
APPLICATION**

05

**TECHNICAL OVERVIEW OF
CONTEXTUALISE**



An abstract geometric network diagram consisting of numerous interconnected nodes and lines, forming a complex web of triangles and polygons. The nodes are represented by small black dots, and the lines are thin, light gray. The overall structure is dense and interconnected, with some nodes having multiple connections. The diagram is positioned in the upper right and lower right areas of the page, partially overlapping the text.

01

THE TOPIC MAPS PARADIGM

Topic maps provide a way to describe complex relationships between abstract concepts and the accompanying real-world (information) resources

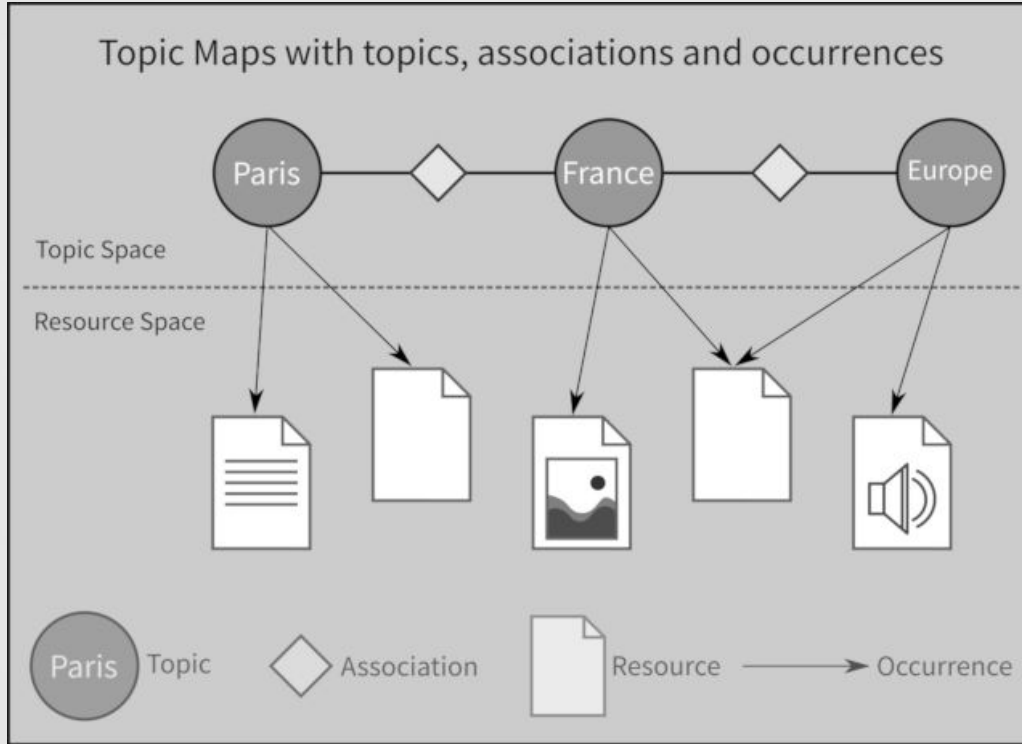
THE TOPIC MAPS PARADIGM

DOMAIN MODEL — AN ASSOCIATIVE GRAPH

- **Topic**: represents an abstract concept
- **Association**: expresses a semantically meaningful relationship between two or more topics
- **Occurrence**: connects an information resource to a topic
- **Scopes** and scope filtering
- **Metadata**



THE TOPIC MAPS PARADIGM





02

USE CASE 1: KNOWLEDGE MANAGEMENT

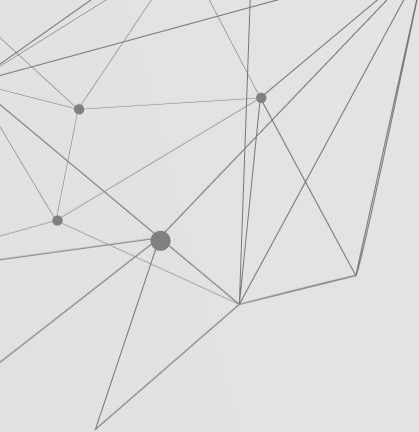
Contextualise is a simple but effective tool particularly suited for organising information-heavy projects and activities consisting of unstructured and widely diverse data and information resources

CONTEXTUALISE: A PERSONAL KNOWLEDGE MANAGEMENT APPLICATION

TOPICS, ASSOCIATIONS AND OCCURRENCES

- Multiple topic maps
- Topics
- Associations
 - ◆ Navigable network graph
 - ◆ Associative tags
 - ◆ Knowledge paths – for easy hierarchical navigation through a topic map
- Occurrences and information resources
 - ◆ Text, images, files, links and videos
 - ◆ glTF-based 3D scenes – with AR and VR support by December 2020





CONTEXTUALISE

LET'S TAKE A CLOSER LOOK





03

USE CASE 2: STORYTELLING AND WORLDBUILDING

Human beings have been telling stories as long as there's been a language to tell them in. We think in stories, remember in stories, and turn just about everything we experience into a story.

SEMANTIC DATA MODEL FOR STORYTELLING AND WORLDBUILDING PURPOSES

EVENTS, PARTICIPANT, OBJECTS (THINGS) AND NARRATIVE RELATIONSHIPS

- Semantic **narrative event**
 - ◆ What? When? Where? Who? Why? How?
- Events are recursive
 - ◆ Sub-events are to events what events are to a narrative: they keep moving the narrative forward, each at their own level
- Relationships
 - ◆ Support for both **spatial and causal relationships**
- **Extending** lower-level topic map model with a higher-level semantic narrative model





04

THE STORYTELLER APPLICATION: A CONTEXTUALISE EXPERIMENT

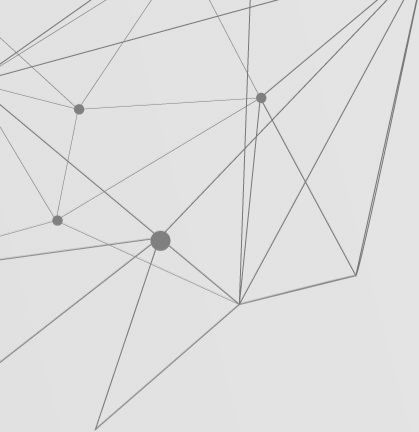
A three.js frontend application talking to a Contextualise/TopicDB backend

STORYTELLER: A THREE.JS-BASED APPLICATION

THREE.JS, WEB SERVICES AND A SEMANTIC GRAPH BACKEND

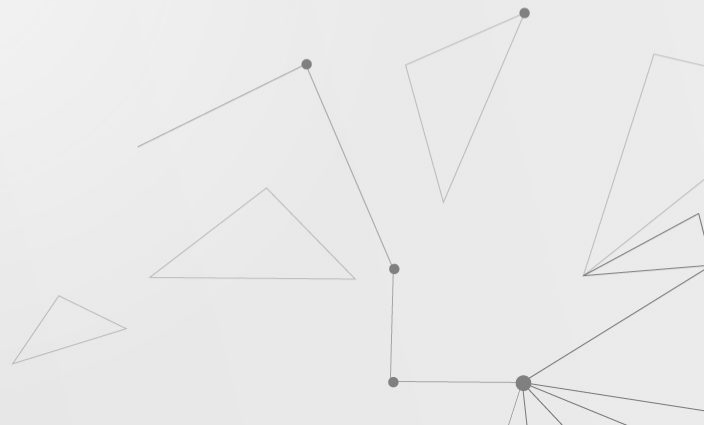
- **Navigation** between narrative events support for sub-events
- Interactive 3D scenes with **Points of Interest**
 - ◆ Participants
 - ◆ Objects (things)
 - ◆ Inter-scene navigation
 - ◆ Tags
- Entity viewer
- **AR** and **VR** support coming in November and December of 2020, respectively





STORYTELLER

LET'S TAKE A CLOSER LOOK





05

TECHNICAL OVERVIEW OF CONTEXTUALISE

Why did Flask make sense for Contextualise? To understand that we need to look at the intersection of Contextualise's architecture and the nature of Flask – hint: it's unopinionated

WHY FLASK?

CONTEXTUALISE'S ARCHITECTURE AND FLASK CHARACTERISTICS

→ Flask

- ◆ Small core
- ◆ Extendable
- ◆ **Unopinionated**

→ Contextualise architecture

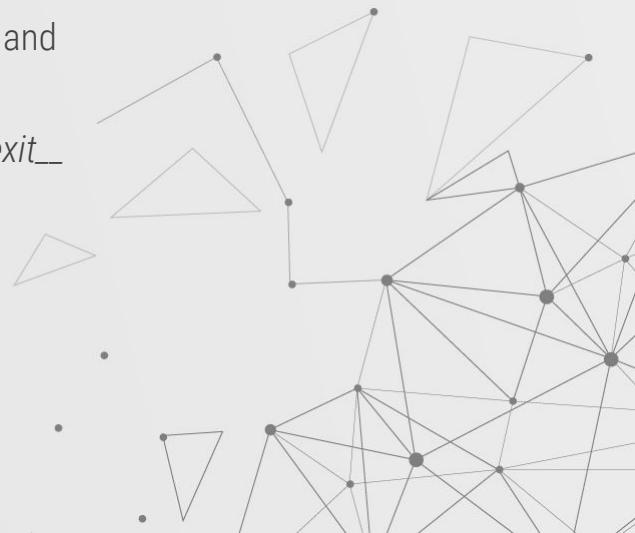
- ◆ Broadly speaking, Contextualise is divided into a web “frontend” on one hand, and a **graph-based backend**, on the other
- ◆ TopicDB, a so-called **topic maps engine** – a variation of the **repository pattern**



THE REPOSITORY PATTERN

ONE OF THE SO-CALLED “ENTERPRISE” PATTERNS

- Mediates between the domain and data mapping layers
- Beneficial for a system with a complex domain model
- Achieves a clean separation and one-way dependency between the domain and data mapping layers
- In Python terms, the repository is a **context manager** with `__enter__` and `__exit__` methods for *open* and *close* (**connection**) semantics

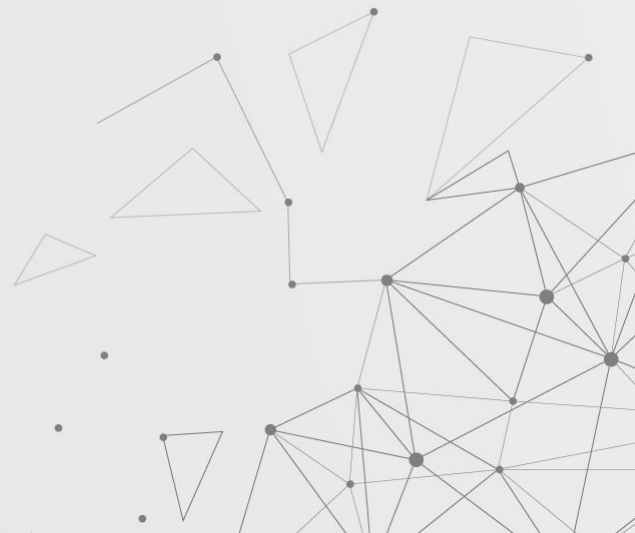


WIRING UP AND USING THE DATA STORE

TOPIC_STORE.PY

```
def get_topic_store():
    if "topicstore" not in g:
        g.topic_store = TopicStore(
            current_app.config["TOPIC_STORE_USER"],
            current_app.config["TOPIC_STORE_PASSWORD"],
            host=current_app.config["TOPIC_STORE_HOST"],
            port=current_app.config["TOPIC_STORE_PORT"],
            dbname=current_app.config["TOPIC_STORE_DBNAME"]
        )
        g.topic_store.open()
    return g.topic_store

def close_topic_store(e=None):
    topic_store = g.pop("topicstore", None)
    if topic_store is not None:
        topic_store.close()
```



WIRING UP AND USING THE DATA STORE

TOPIC_STORE.PY (CONTINUED)

```
def init_app(app):  
    app.teardown_appcontext(close_topic_store)
```

__INIT__.PY

```
from contextualise import topic_store  
  
topic_store.init_app(app)
```



WIRING UP AND USING THE DATA STORE

VIDEO.PY (BLUEPRINT)

```
@bp.route("/videos/<map_idenfier>/<topic_idenfier>")
@login_required
def index(map_idenfier, topic_idenfier):
    topic_store = get_topic_store()
    topic_map = topic_store.get_topic_map(map_idenfier, current_user.id)
    if topic_map is None:
        abort(404)
    if not topic_map.owner and topic_map.collaboration_mode is not CollaborationMode.EDIT:
        abort(403)
    topic = topic_store.get_topic(map_idenfier, topic_idenfier)
    if topic is None:
        abort(404)
```





THANKS! ANY QUESTIONS?

Does anyone have any questions?

info@contextualise.dev
<https://contextualise.dev>
<https://brettkromkamp.com>
<https://github.com/brettkromkamp>

