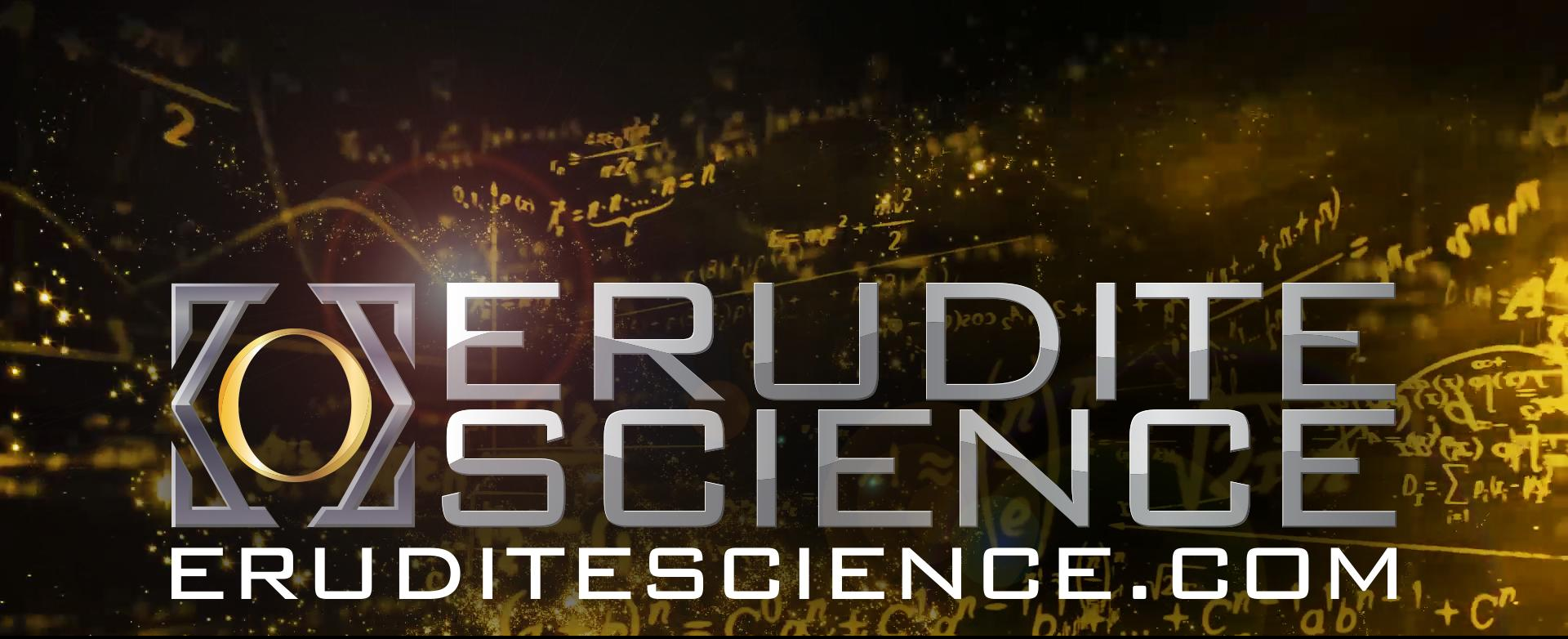


Matrix: open standard for persistent communication built on python

Our Problem

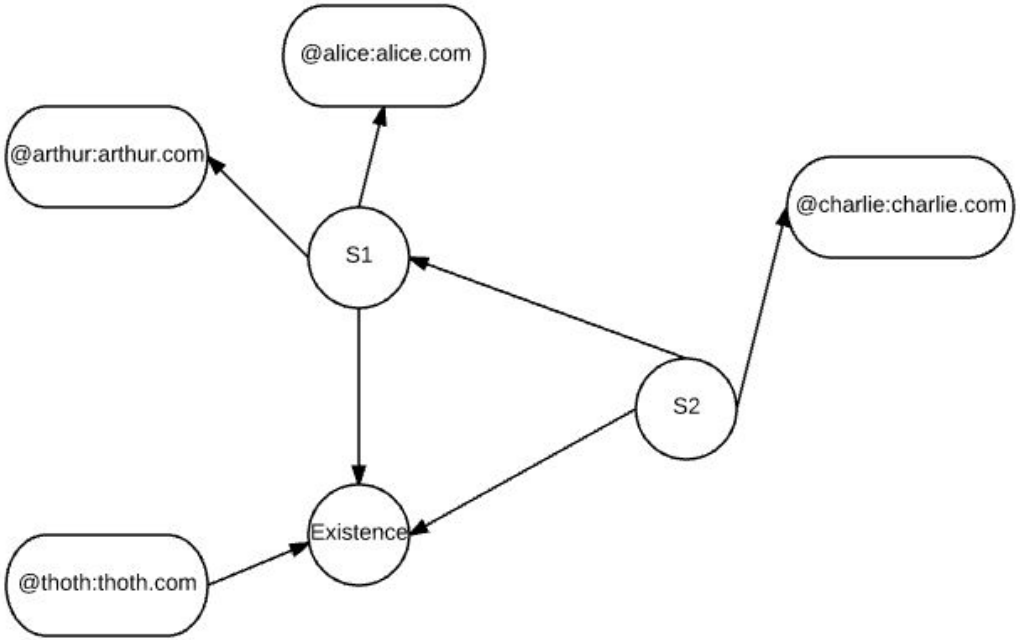
- Pupil: a peer-to-peer tutoring chatbot
- Liquid Crowd: messaging system on crowd equity platform

- Requirements:
  - Security (Slack, Mattermost)
  - Message interception for reply automation (chatbot augmentation)

Our Solution

CHATBOOT

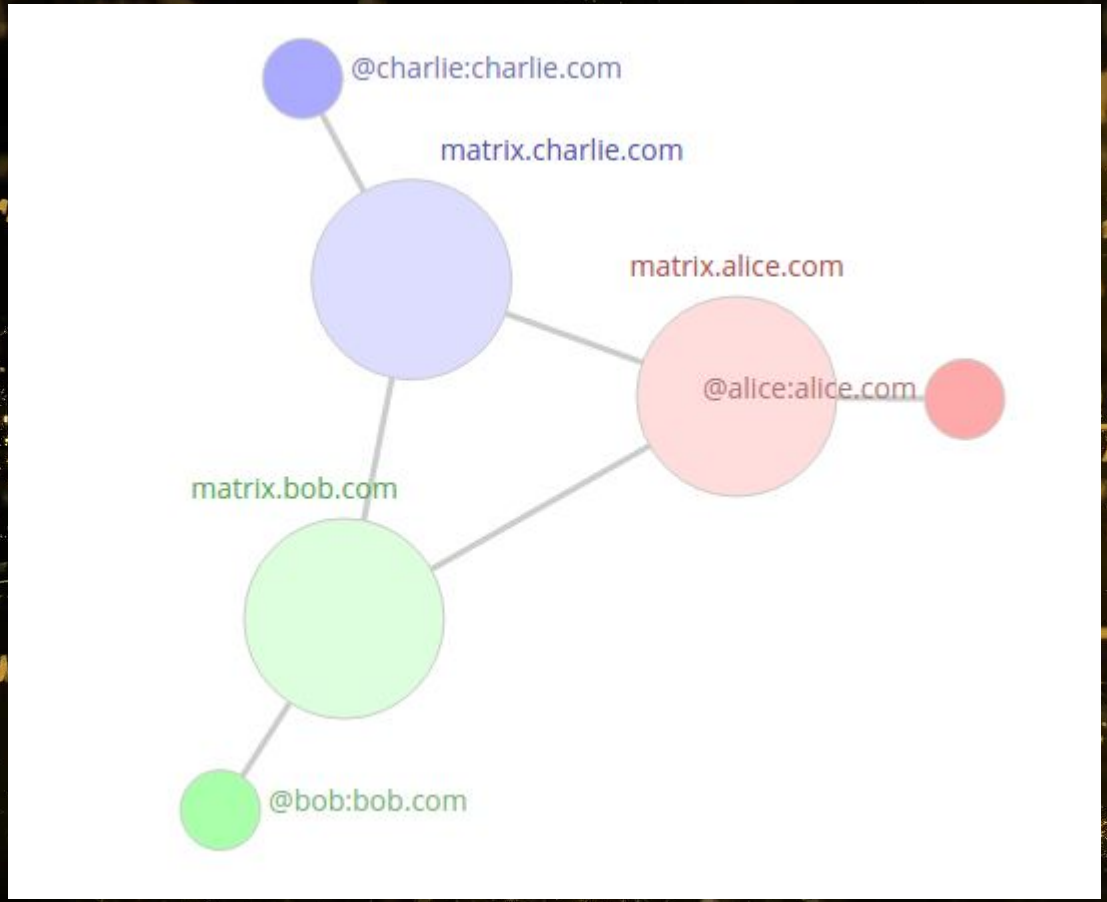
Alice: Hello Charlie, I am having problem solving 5+2, I get 3. Could you help me out?



Thoth:

1) Hello Alice, sure let me help you. First, can you show me your steps?
2) Hi Alice, it will be my pleasure.
3) …

Charlie: 1


@charlie:charlie.com
matrix.charlie.com
matrix.alice.com
@alice:alice.com
matrix.bob.com
@bob:bob.com

Synapse is written in python 2.7 but some of the libraries are written in C.

Matrix APIs

The following APIs are documented in this specification:

API	Version	Description
Client-Server API	r0.2.0	Interaction between clients and servers
Server-Server API	unstable	Federation between servers
Application Service API	unstable	Privileged server plugins
Identity Service API	unstable	Mapping of third party IDs to Matrix IDs
Push Gateway API	unstable	Push notifications for Matrix events

Create Room

```
curl -XPOST -d '{"room_alias_name":"tutorial"}'
"https://localhost:8448/_matrix/client/r0/createRoom?access_token=YOUR_ACCESS_TOKEN"

{
    "room_alias": "#tutorial:localhost",
    "room_id": "!asfLdzLnOdGRkdPZWu:localhost"
}
```

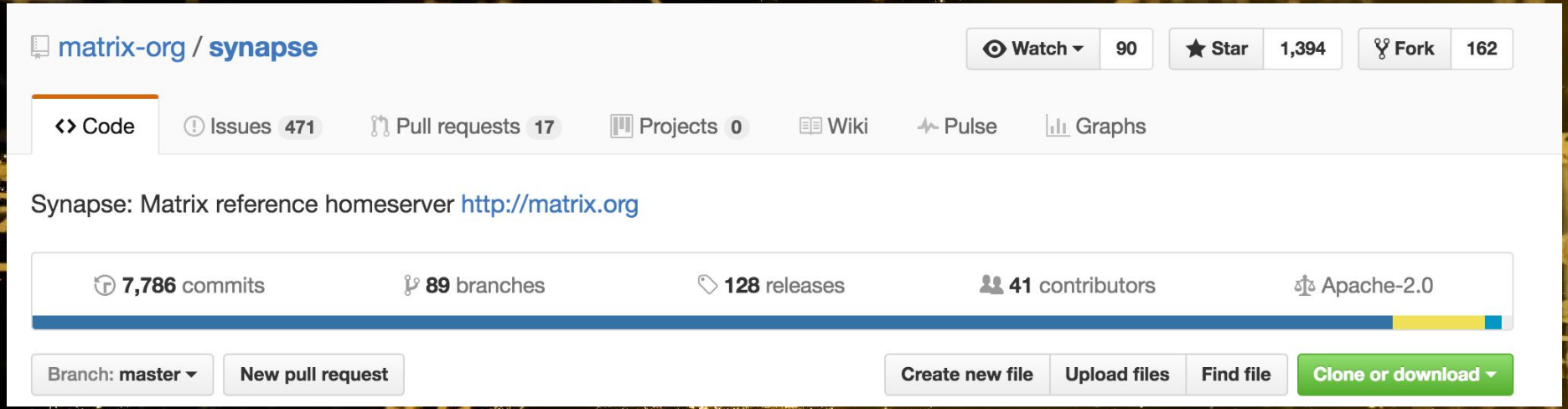
Dependencies:

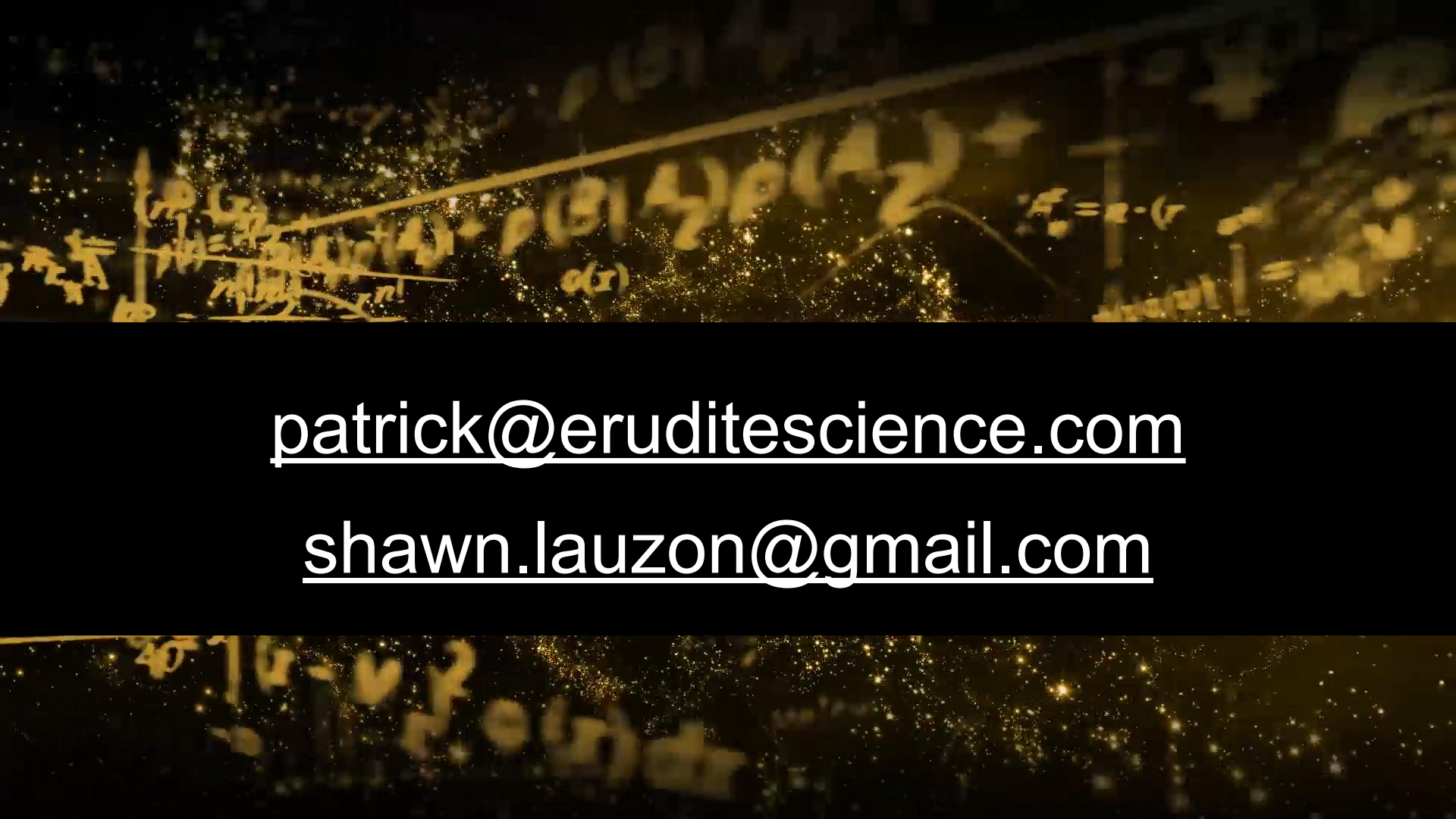
- **Synapse**: Reference Matrix server in Python
- **Hello-Matrix**: Simple bot using Javascript SDK to connect to external services

Demo:

1. Create Synapse demo federation of 3 servers (running on ENTER URLS HERE)
2. Configure and run hello-matrix bot
3. Start 2 clients for our users
4. Invite hello-matix bot
5. Send messages between users, see in log
6. Use !calculate to invoke Wolfram Alpha

About Matrix





patrick@eruditescience.com

shawn.lauzon@gmail.com