

An abstract graphic on the left side of the slide, consisting of white lines and circles on a blue gradient background, resembling a circuit board or data flow diagram.

BIG BROTHER

1. MILESTONE

DATABASE

- Ressources
- GridFS
- MongoDB
- Database structure
- 1. Milestone

DATABASE – RESSOURCES

- **Getting Started:**

- Introduction into how it works:
<https://www.mongodb.com/docs/manual/tutorial/getting-started/>
- Setting atlas up: <https://www.mongodb.com/docs/atlas/getting-started/>

- **With Python (and pymongo):**

- Introductory: https://www.youtube.com/watch?v=rE_bJI2GAY8
- Reading more about it:
 - Very short tutorial: <https://www.mongodb.com/docs/drivers/pymongo/>
 - <https://pymongo.readthedocs.io/en/stable/>
-> <https://pymongo.readthedocs.io/en/stable/tutorial.html>

- **Design:**

- <https://www.youtube.com/watch?v=leNCfU5SYR8>
- Schema Design Patterns: <https://www.mongodb.com/blog/post/building-with-patterns-a-summary>
- How not to design - Anti-patterns: <https://www.youtube.com/watch?v=8CZs-0it9r4&list=PL4RCxklHWZ9uluV0YBxeuwpEa0FWdmCRy>

GRIDFS

- MongoDB doesn't allow you to store files that are larger than 16MB
- In order to store them you would need GridFs
- Breaking up files into
 - fs.files: Contains data about those files
 - fs.chunks: Contains data about chunks
- Makes loading videos easier

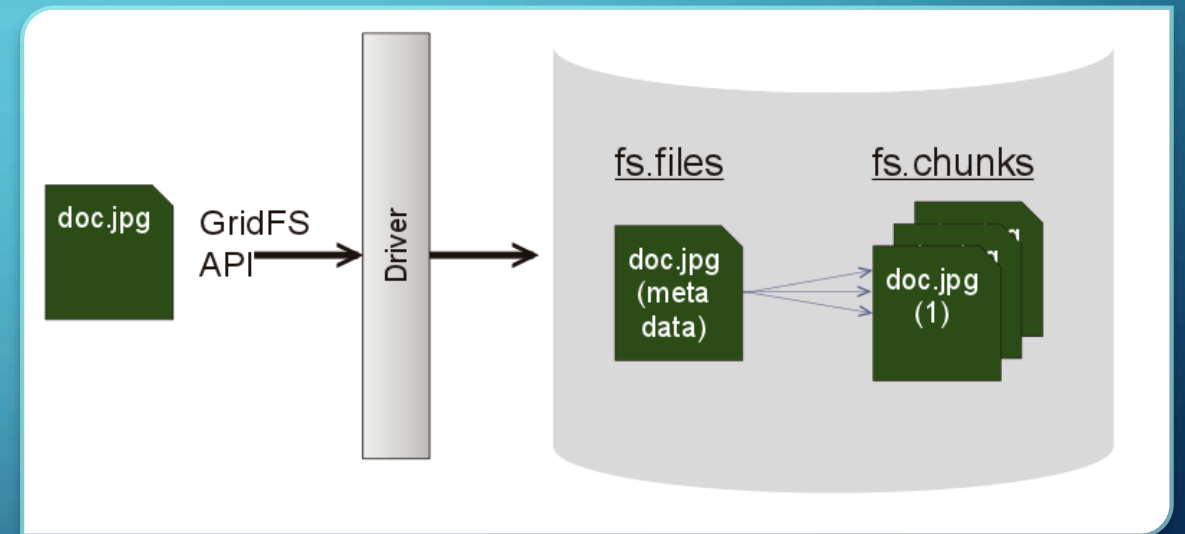


Abb. 1: [gridfs1.png \(797×365\)](#)

GRIDFS – RESSOURCES

- Overview: <https://www.youtube.com/watch?v=GDUbWNJLPnc>
- Documentation: <https://www.mongodb.com/docs/manual/core/gridfs/>
- With pymongo:
 - Introduction: <https://pymongo.readthedocs.io/en/stable/examples/gridfs.html>
 - Read more: <https://pymongo.readthedocs.io/en/stable/api/gridfs/index.html#module-gridfs>

MONGODB

- Not classical relational model
- Adaptable to queries
- More flexible
- More manageable

Database

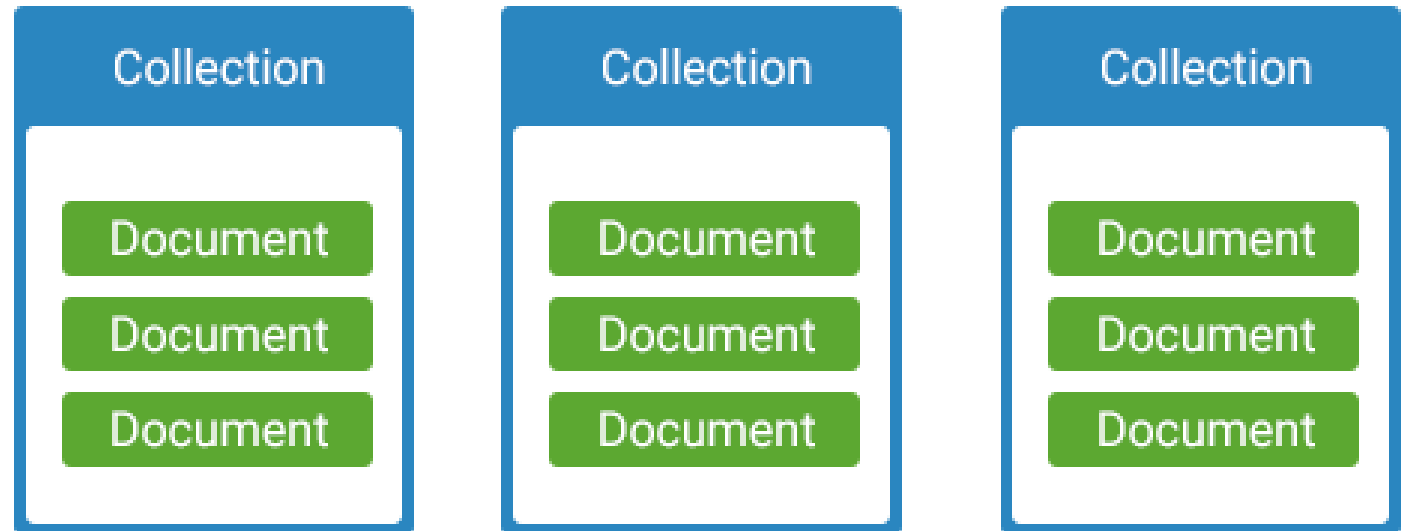


Abb. 2: <https://www.koderhq.com/img/mongodb/collections.png>

Database – Big Brother

User***

- _id: UUID*
- username: String
- user_enc: String
- is_admin: Bool

Login_attempt***

- _id: UUID*
- user_id: UUID*
- date: Date
- login_suc: String
- success_res: String

Ressource_context**

- _id: UUID*
- username: String
- res_id: [UUID*]

Ressource**

- _id: UUID*
- user_id: UUID*
- date: Date
- res_cxt_id: [UUID*]

* UUIDs are being stored as Strings

** Ressource and Ressource_context: used for training data

*** User and Login_attempt is used for user logins

1. MILESTONE

- **Done:**
 - Created scheme for database
 - Converted Database into MongoDB
 - Setup and write documentation for Database management API
- **Next milestone:**
 - Discussing small scheme changes to load documents more efficient
 - Finish implementing scheme
 - Implementing further queries
 - Help frontend with fixing issues connected to the database
 - Create user-groups on DB-side to rights

LOGIK

- Python Module (until now)
- 1. Milestone
- TODO

PYTHON MODULE (UNTIL NOW)

- OpenCV (mainly)
- numpy (math)
- mediapipe & tensorflow (still some problems we need to deal with) (machine learning)

1. MILESTONE

- Video resource
- Live face recognition
- Palm detection
- Gesture detection and other functions related with finger movement



TODO

- more useful functions/improvement of current algorithms
- determine and achieve the highest possible efficiency of the algorithms (for example, excluding the possibility to abuse the face recognition)
- Enabling the program to work better in poor lighting conditions (providing user with information on better camera positioning)
- Cooperation (parallel) “Live face recognition”- and “Gesture detection” - programs

FRONTEND

- Removed obsolete or unused requirements
- Python versions customized
- Pip installations all performed
- Code was intensively considered and analyzed
 - Make Code less redundant and clear

FRONTEND – NEW MILESTONE



- Make Code less redundant and clear
- resolve interface issues
- completeness and functionality of the website