

# LOCAL EGA

## DOCUMENTATION

The screenshot shows a web browser window with the URL `localega.readthedocs.io/en/latest/`. The page title is "NBIS - Local EGA". The left sidebar contains a navigation menu with the following items: "Local EGA", "latest", "Search docs", "Installation", "Configuration", "Logging", "Bootstrap", "Inbox", "Ingestion", "Encryption", "Database", "CEGA from/to LEGA", "API documentation", "Contributing", "Policies", and a promotional banner for Read the Docs. The main content area is titled "NBIS - Local EGA" and includes a link to "Edit on GitHub". The text explains that throughout the documentation, Central EGA is referred to as `CEGA`, `CentralEGA`, or `any Local EGA instance as LEGA`, or `LocalEGA`. When two or more Local EGA instances are involved, the notation `LEGA<1>` is used for Local EGA instance `<1>`. The section "NBIS - Local EGA" states that the project is divided into several microservices, which are listed in a table:

DB	A Postgres database with appropriate schema	□
MQ	A RabbitMQ message broker with appropriate accounts, exchanges, queues and bindings	□
INBOX	SFTP server, acting as a dropbox, where user credentials are in the Central EGA	□
KEYSERVER	Handles the encryption/decryption keys	□
WORKERS	Connect to the keyserver (via SSL) and do the actual re-encryption task	□
VAULT	Moves files from the staging area to the vault storage, including a verification step afterwards.	□

The workflow consists of two ordered parts:

The user first logs onto the Local EGA's inbox and uploads its files. He/She then goes to the Central EGA's interface to prepare a submission. Upon completion, the files are ingested into the vault and become searchable by the Central EGA's engine.

More concretely, Central EGA contains a database of users. The Central EGA's ID is used to authenticate the user against either their EGA password or an RSA key.

# LOCAL EGA TECHNOLOGIES



DOCKER