

# Aufbau

- [Worker-System](#)

# Worker-System

## Überblick

Dokumente in SQUEEZE werden mit einer Pipeline verarbeitet, einer Sammlung verschiedener Einzelschritte wie z.B. Bildaufbereitung und Texterkennung.

Diese einzelnen Schritte werden dabei von Worker-Prozessen ausgeführt, die unabhängig der API arbeiten.

Ein Worker-Manager koordiniert die Job-Ausgabe und sorgt für eine faire Verteilung bzw. Priorisierung.

Kommt ein neues Dokument in den Verarbeitungsprozess, wird von der SQUEEZE-API ein neuer Job für das Dokument an den Worker-Manager übermittelt. Der nächste freie Worker fordert einen Job vom Worker-Manager an und führt den nächsten anstehenden Pipeline-Schritt aus.

Pro Pipeline-Schritt wird zuerst vom jeweiligen Worker ein Archiv mit den erforderlichen Daten von der API heruntergeladen und in sein lokales Dateisystem übertragen.

Danach beginnt die eigentliche Verarbeitung, z.B. die Bildaufbereitung oder Extraktion.

Am Ende des Vorgangs wird das Result wieder als Archiv über HTTP zur API hochgeladen.

## Aufbau des Workers

Der Worker und Worker-Manager sind Teil eines einzigen ausführbaren Programmes. Für die Manager-Funktion wird die Executable mit dem zusätzlichen Parameter ``manager`` gestartet.

Worker und Worker-Manager sind in Go geschrieben, die eigentliche Ausführung des Pipeline-Schrittes wird jedoch durch ein PHP-Script (`bridge.php`) bewerkstelligt, was den SQUEEZE-Core lädt und den Schritt ausführt.

