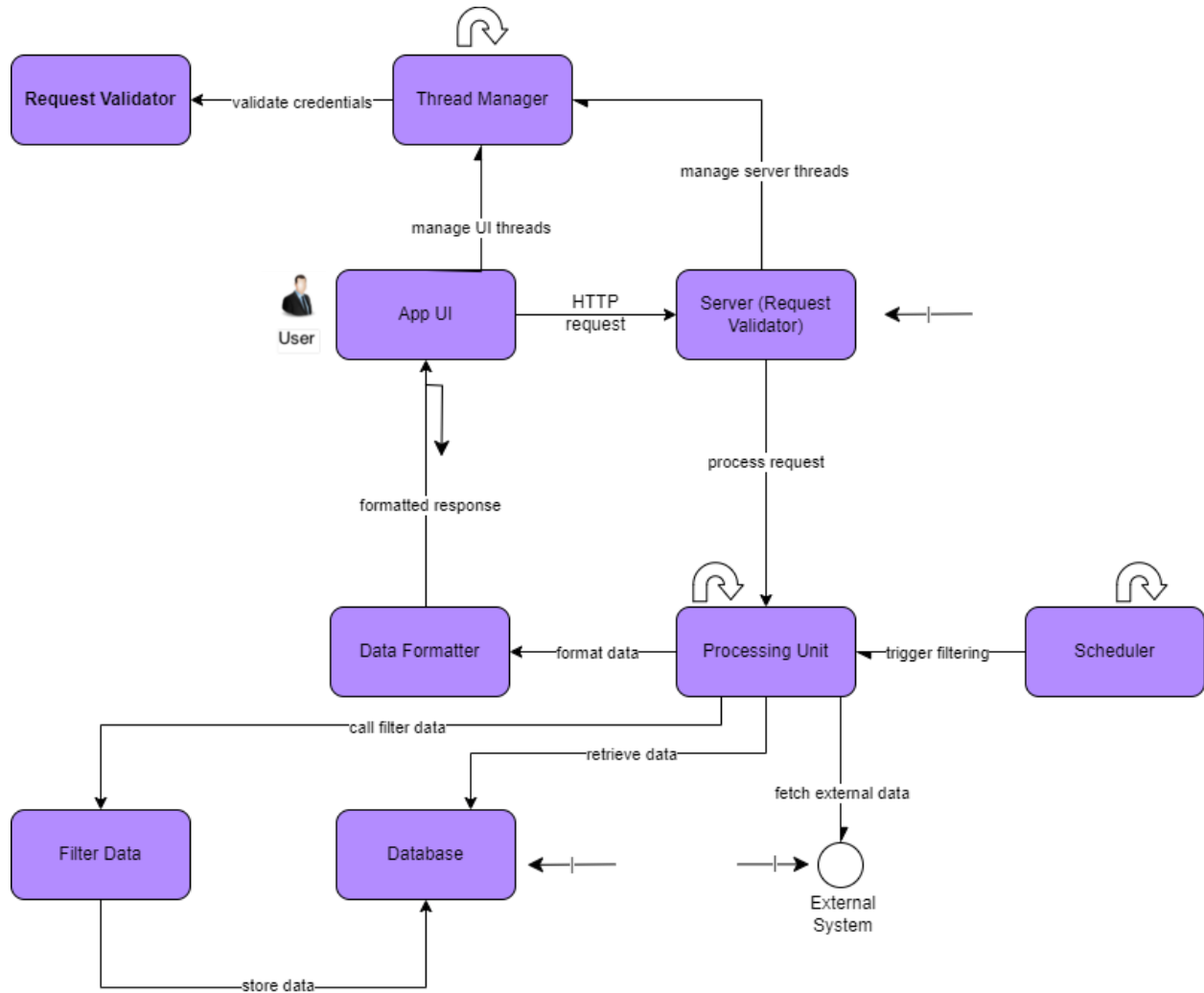


ИЗВРШНА АРХИТЕКТУРА



1. Кориснички интерфејс и иницијализација:

- Корисникот преку **App UI** иницира барање.
- Барањето се валидира од **Request Validator** пред да биде испратено до серверот преку **Thread Manager**, кој управува со нишки.

2. Обработка на барањето:

- **Server (Request Validator)** прима и валидира барања, потоа ги предава до **Processing Unit** за обработка.

- **Processing Unit** може да побара податоци од **Database** или да се поврзе со **External System** за дополнителни информации.

3. Форматирање на податоци:

- **Data Formatter** подготвува одговори во соодветен формат за враќање кон корисникот преку **App UI**.

4. Филтрирање на податоци:

- **Filter Data** обработува податоци чекор по чекор, со поддршка од **Scheduler** за автоматизација и иницирање.

5. Надворешна интеграција:

- **External System** обезбедува податоци, кои се обработуваат преку **Processing Unit** и се зачувуваат во **Database**.

6. Автоматизација:

- **Scheduler** иницира задачи за обработка во **Data Filter**, односно превземање на податоци од **External System**.

7. Управување со нишки:

- **Thread Manager** ги управува процесите за сервер и кориснички интерфејс, обезбедувајќи паралелизам и оптимални перформанси.

Дијаграмот ја истакнува **динамичката структура** на системот, како и интеракцијата меѓу компонентите за перформанси, скалабилност и интеграција.