Laboratory Session #02

Distributed Systems Programming

Daniele Bringhenti

gRPC Features

- gRPC (https://grpc.io/) is a modern open-source **high performance** framework implementing the remote procedure call (***RPC***) paradigm.
- The main features of gRPC are:

1) simple service definition (Protocol Buffer);
2) high performance and scalability;
3) bi-directional streaming support;
4) multi-language and multi-platform.

gRPC Features

- gRPC (https://grpc.io/) is a modern open-source **high performance** framework implementing the remote procedure call (***RPC***) paradigm.
- The main features of gRPC are:
  1) simple service definition (Protocol Buffer);
  2) high performance and scalability;
  3) bi-directional streaming support;
  4) multi-language and multi-platform.

gRPC is suitable for **Machine-To-Machine (M2M)** communications.

Topics of the Laboratory Session

Laboratory Session #02 covers the following activities:

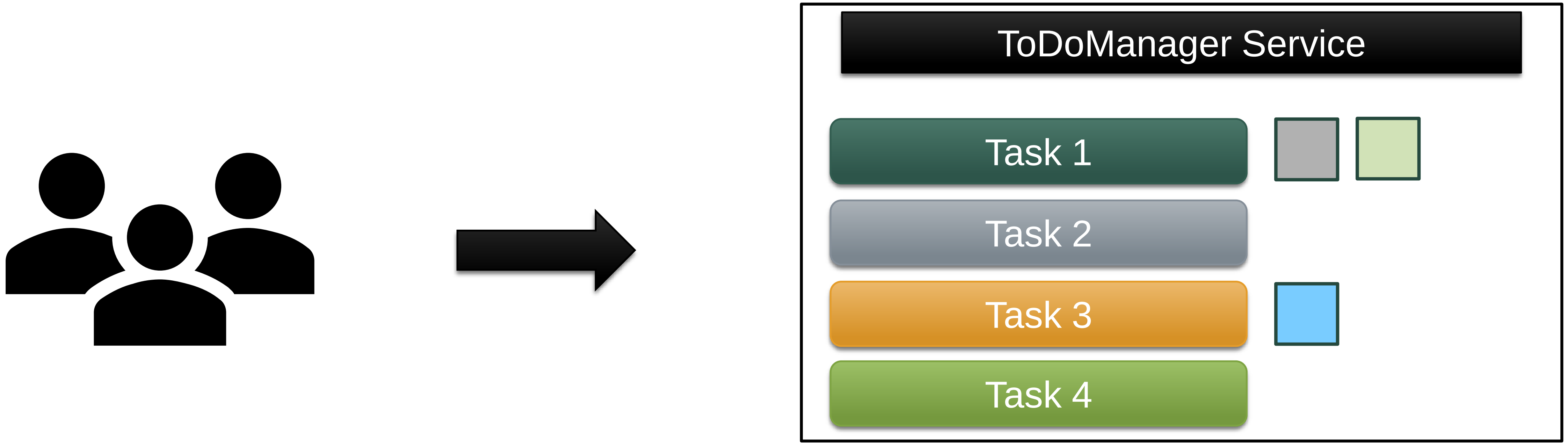
{ REST }

Definition of new **REST APIs** exposed by the ToDoManager service for the management of **images**

gRPC

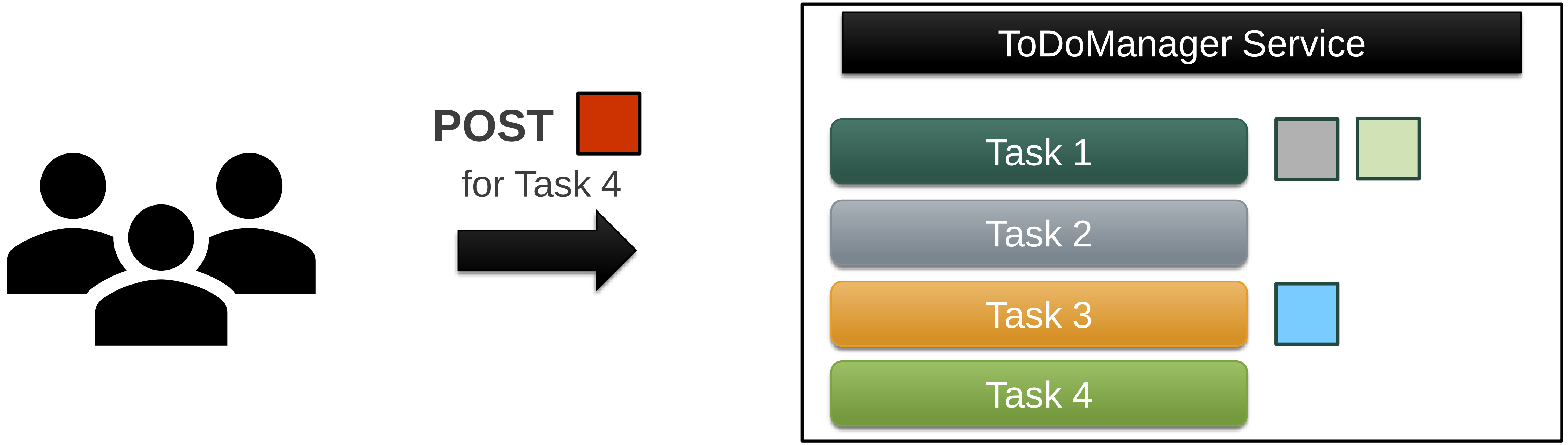
Integration of a **gRPC client** functionality in the implementation of the ToDoManager service

Image Management in the ToDoManager service (I)



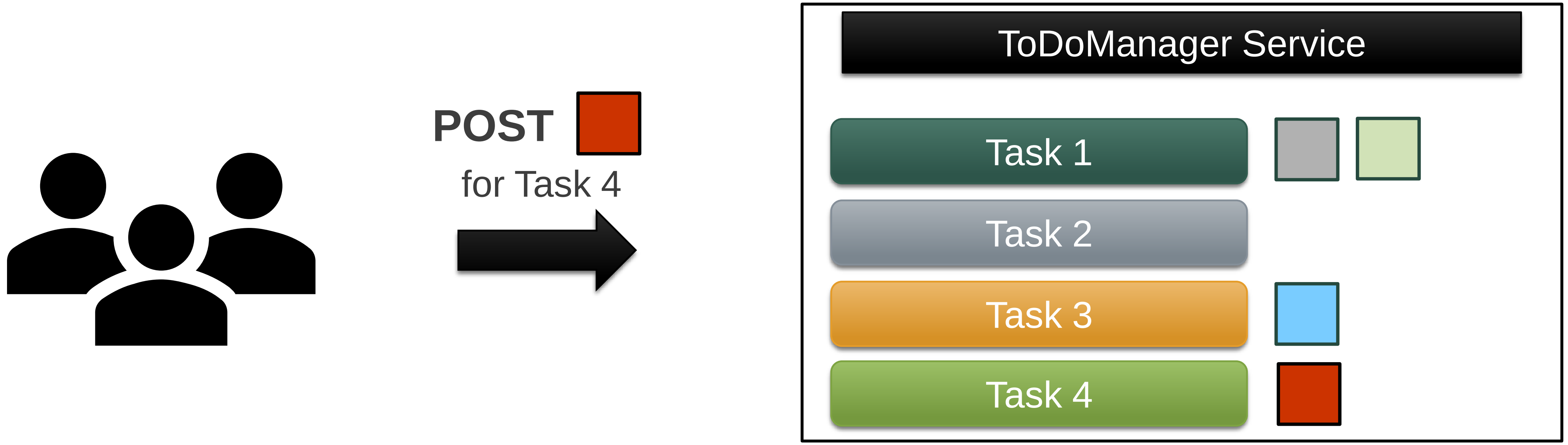
- A user can associate **multiple** images to each task.
- The allowed **media types** are: PNG, JPEG, GIF.
- The service stores the image with its media type.

Image Management in the ToDoManager service (I)



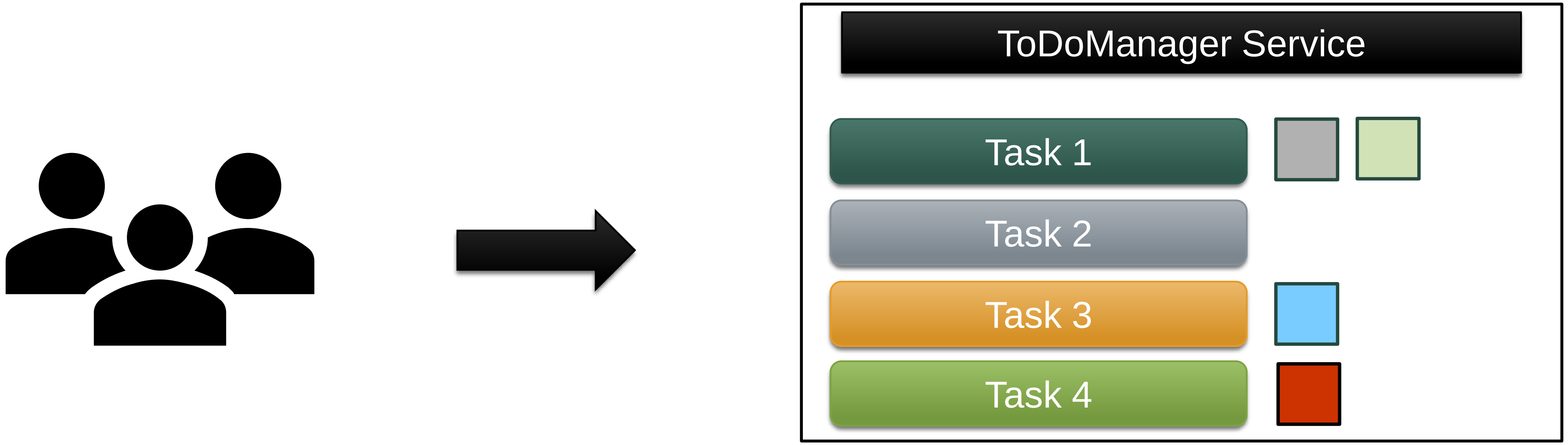
- A user can associate **multiple** images to each task.
- The allowed **media types** are: PNG, JPEG, GIF.
- The service stores the image with its media type.

Image Management in the ToDoManager service (I)



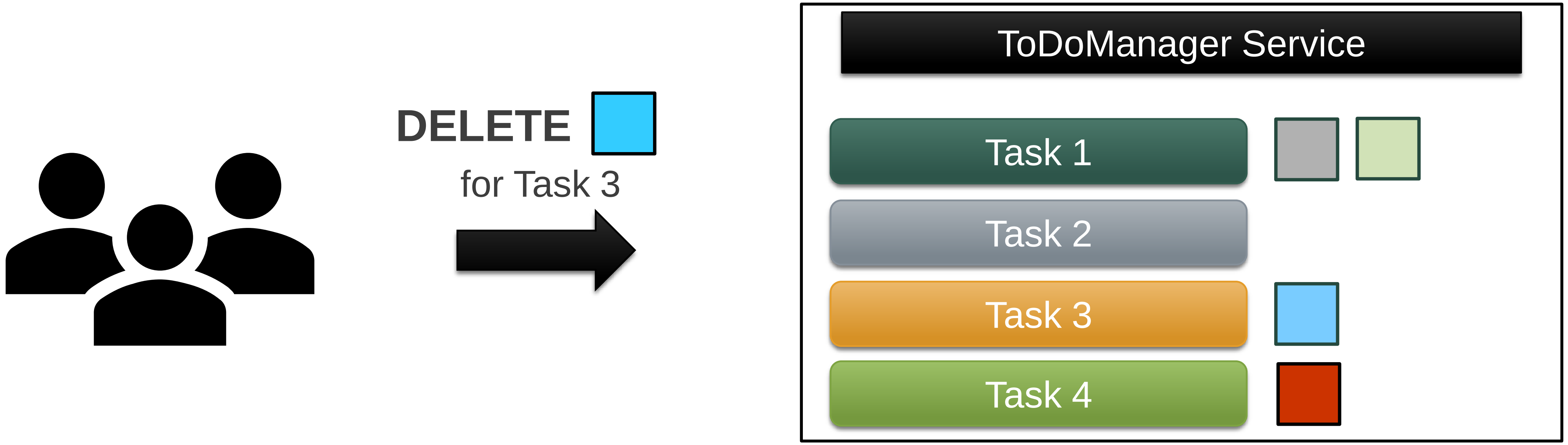
- A user can associate **multiple** images to each task.
- The allowed **media types** are: PNG, JPEG, GIF.
- The service stores the image with its media type.

Image Management in the ToDoManager service (II)



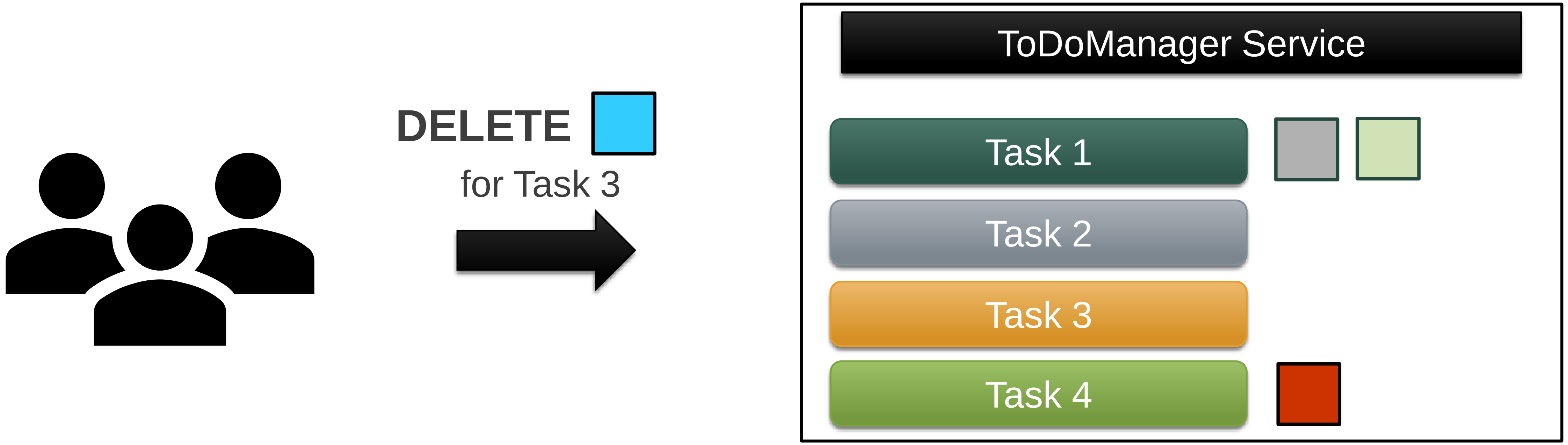
- A user can **delete** an image associated to a task.
- The image is not saved anymore server-side.

Image Management in the ToDoManager service (II)



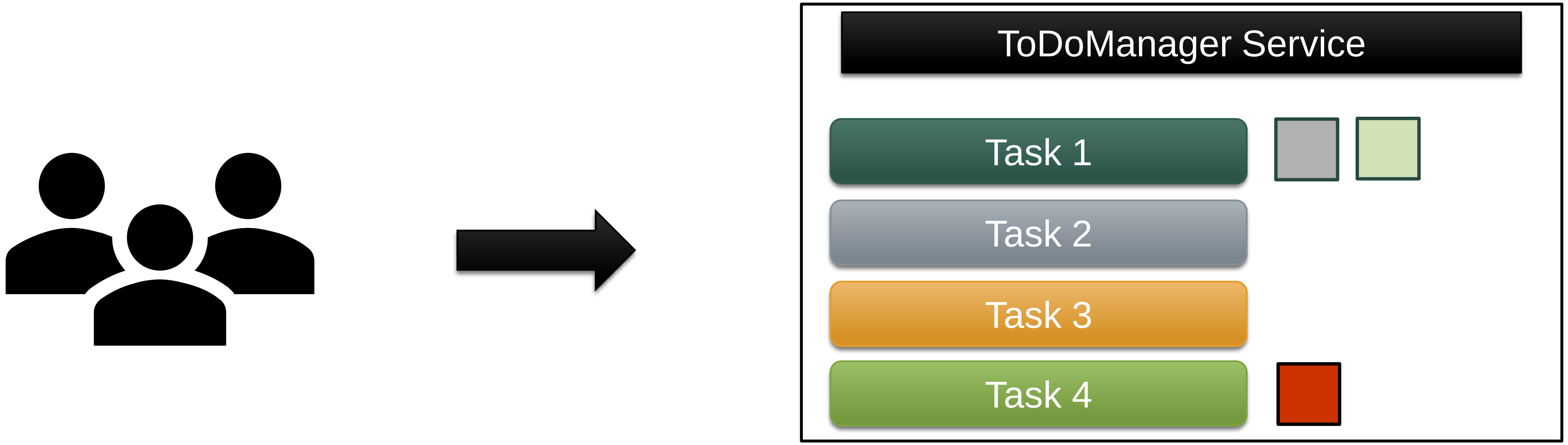
- A user can **delete** an image associated to a task.
- The image is not saved anymore server-side.

Image Management in the ToDoManager service (II)



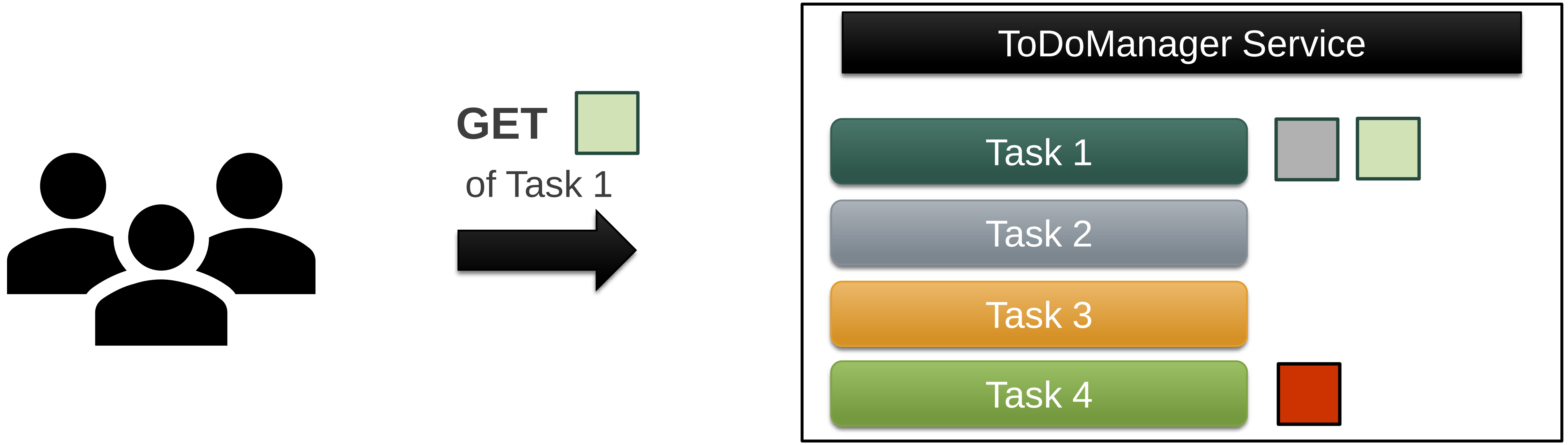
- A user can **delete** an image associated to a task.
- The image is not saved anymore server-side.

Image Management in the ToDoManager service (III)



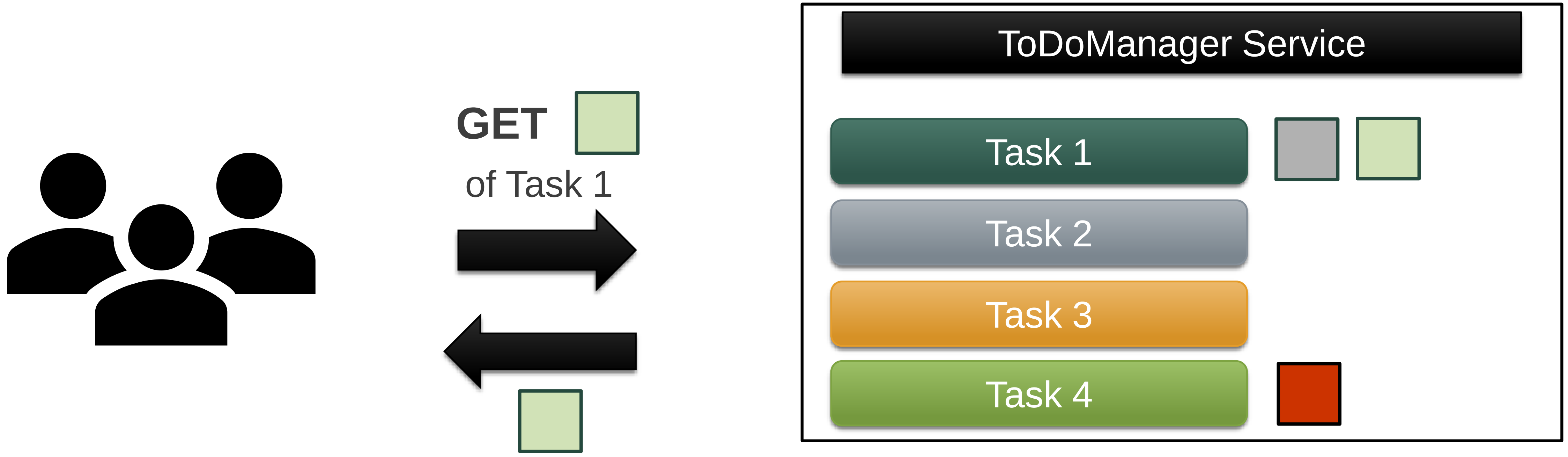
- A user can **retrieve** an image associated to a task.
- The **Accept** header of the HTTP request specifies the requested media type.
- Three media types are supported: image/**png**, image/**jpg**, and image/**gif**. In case the user requests another media type, the operation fails.

Image Management in the ToDoManager service (III)



- A user can **retrieve** an image associated to a task.
- The **Accept** header of the HTTP request specifies the requested media type.
- Three media types are supported: image/**png**, image/**jpg**, and image/**gif**. In case the user requests another media type, the operation fails.

Image Management in the ToDoManager service (III)



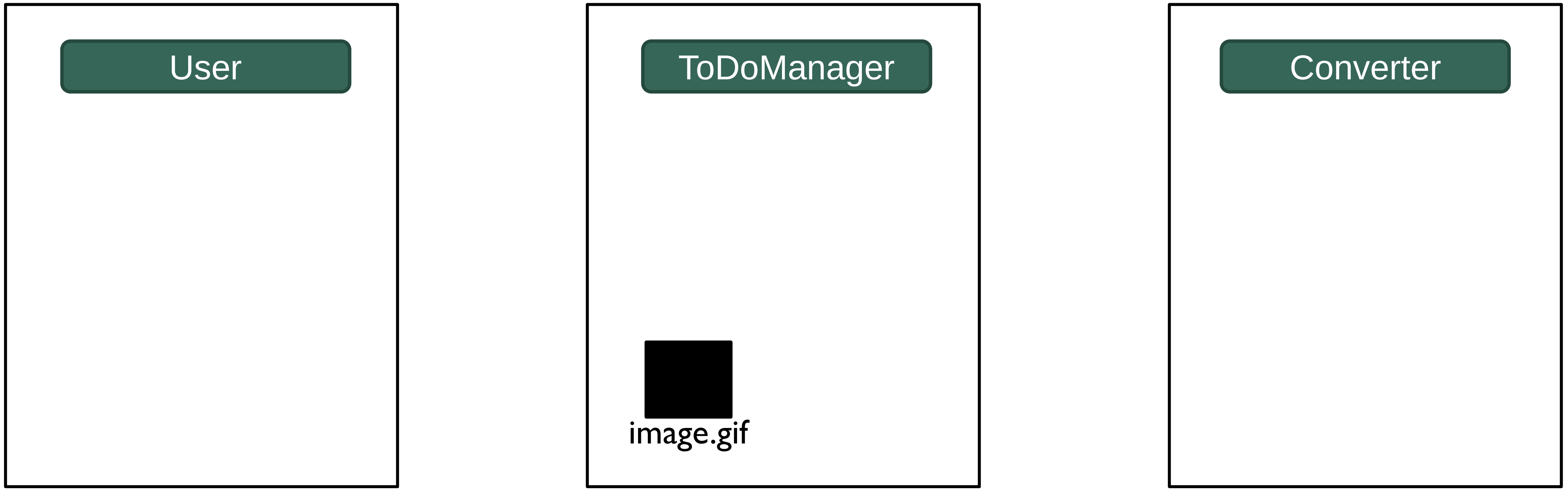
- A user can **retrieve** an image associated to a task.
- The **Accept** header of the HTTP request specifies the requested media type.
- Three media types are supported: image/**png**, image/**jpg**, and image/**gif**. In case the user requests another media type, the operation fails.

Image Media Type Conversion

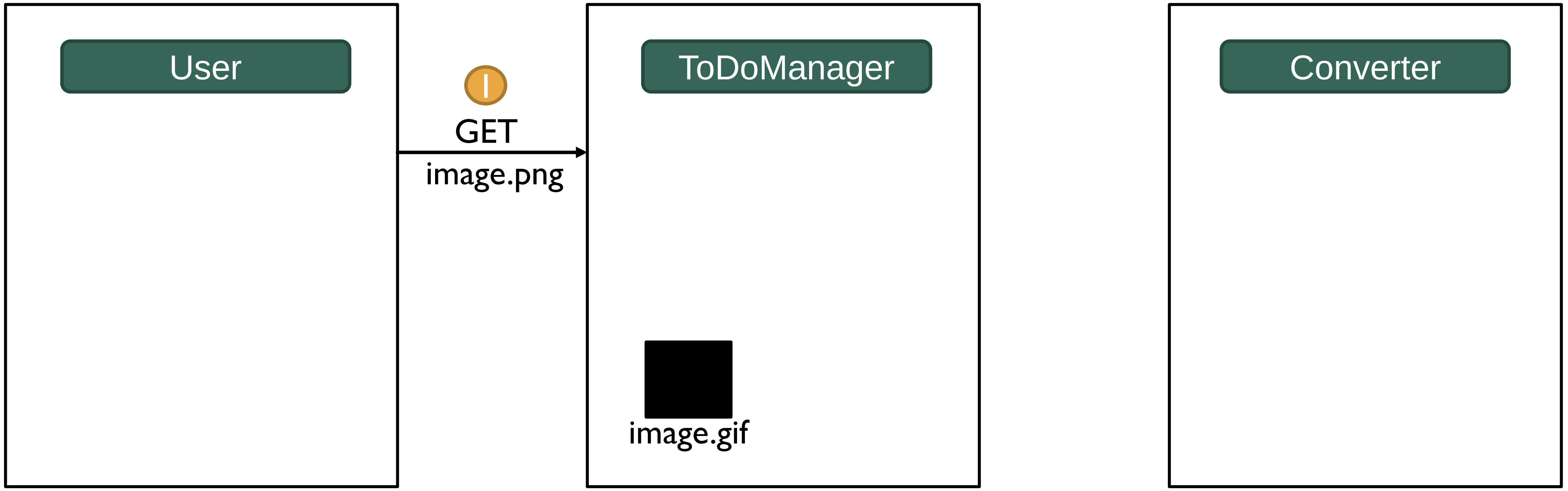
But… what happens if a requested image is saved in a **different media type** than the requested one?

- The *ToDoManager* service interacts with another service, called *Converter* service, **delegating** the operation of media type conversion.
- The *ToDoManager* service creates a **gRPC channel** towards the *Converter* service (i.e., the *ToDoManager* service covers the role of gRPC client, the *Converter* service covers the role of gRPC server).

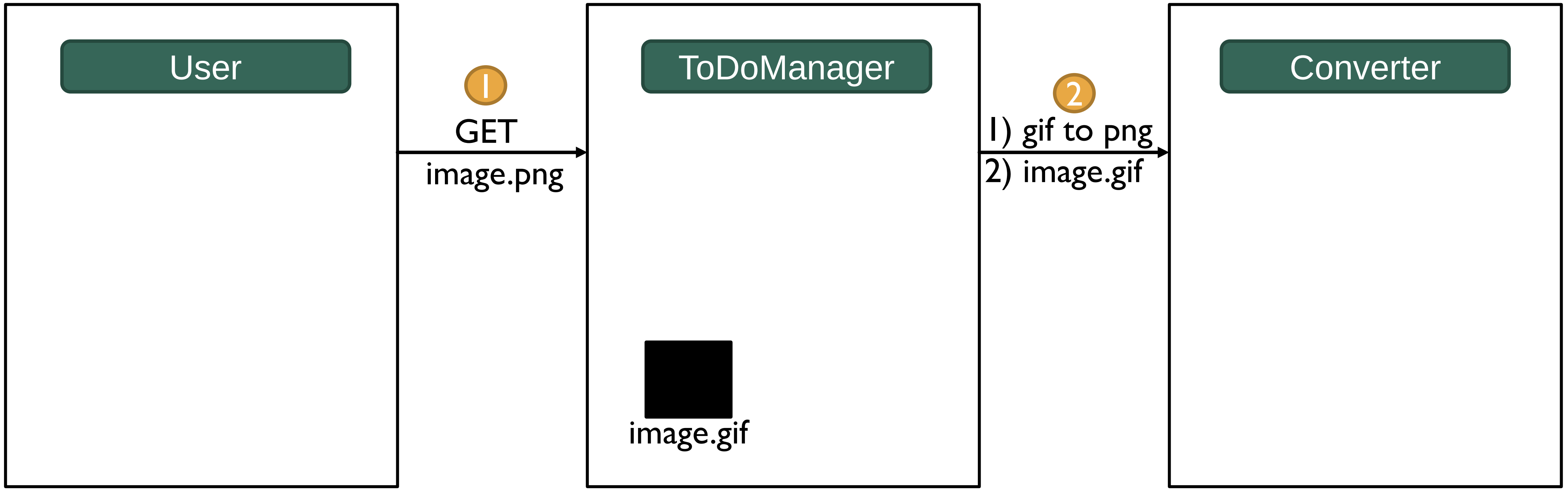
gRPC Communication



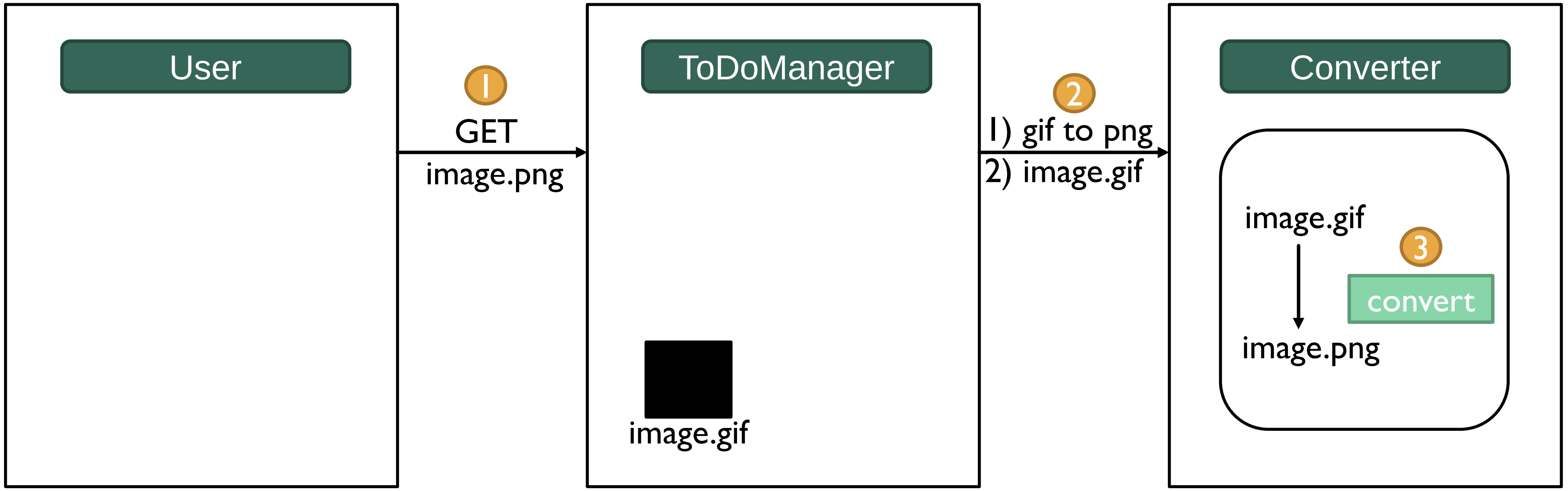
gRPC Communication



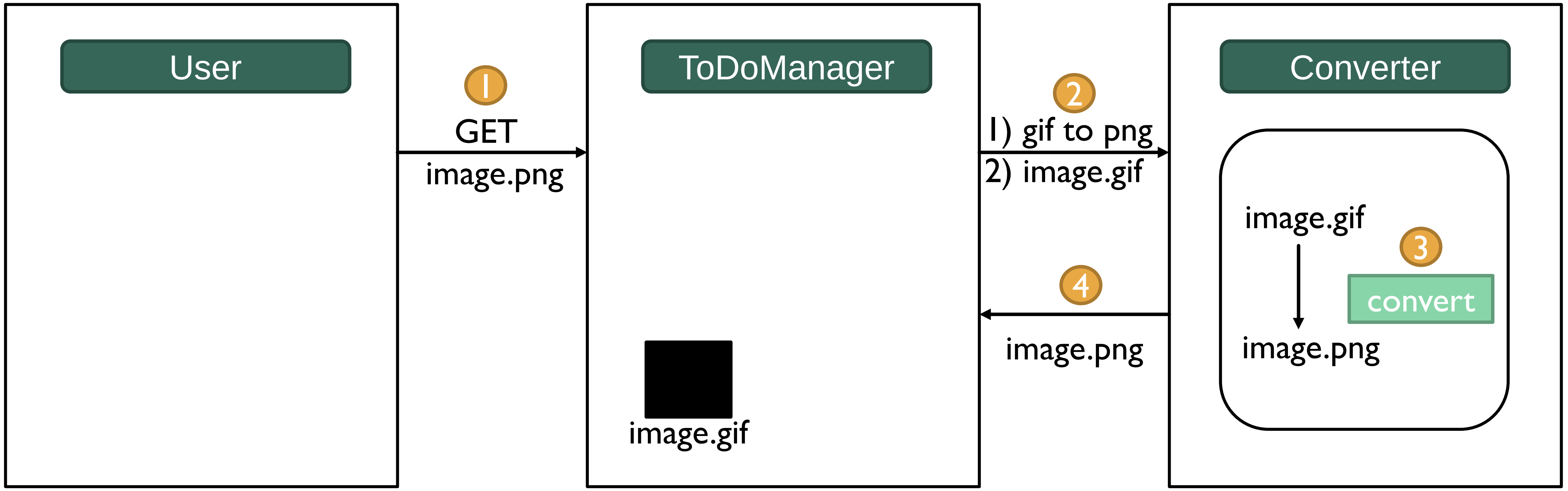
gRPC Communication



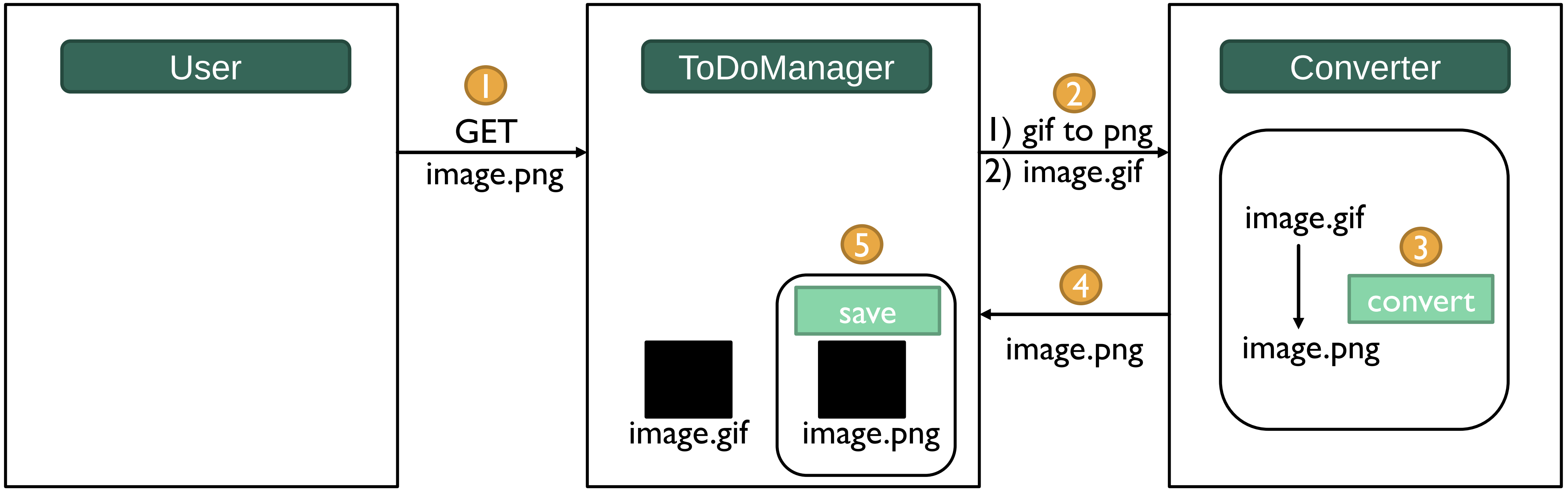
gRPC Communication



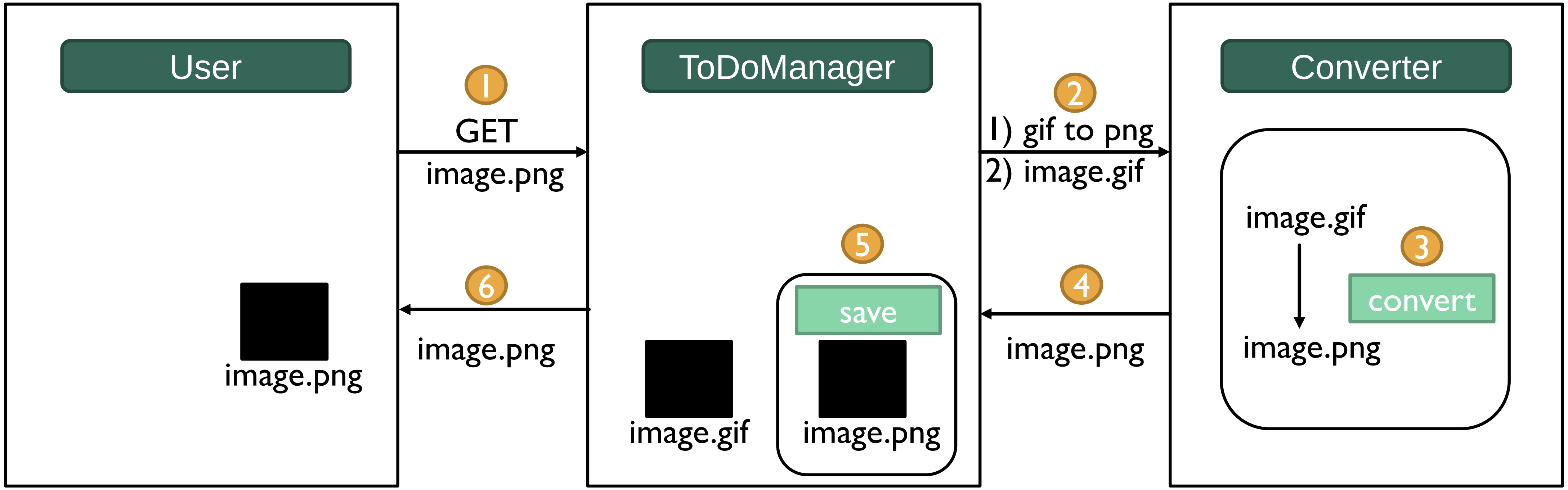
gRPC Communication



gRPC Communication



gRPC Communication



How to manage image files?

Image files must be managed in:

- **Postman**
  - send/receive images in HTTP requests/responses
- *ToDoManager* service (**Node.js**)
  - send/receive images to/from Postman and the *Converter* service
  - locally save images
- *Converter* service (**Java**)
  - send/receive images to/from the *ToDoManager* service
  - convert media types

How to manage image files?

Image files must be managed in:

- **Postman**
  - send/receive images in HTTP requests/responses
- *ToDoManager* service (**Node.js**)
  - send/receive images to/from Postman and the *Converter* service
  - locally save images
- *Converter* service (**Java**)
  - send/receive images to/from the *ToDoManager* service
  - convert media types

Let’s see some tips for the image management!

Thanks for your attention!

Daniele Bringhenti
daniele.bringhenti@polito.it