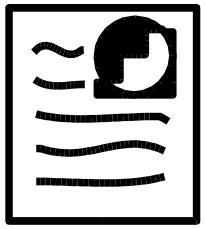
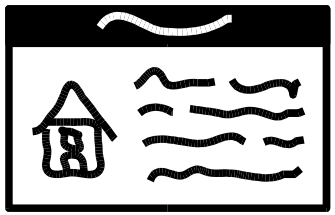
StepTalk Scripting Introduction



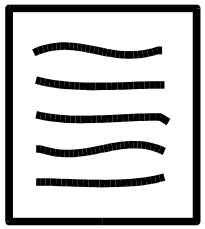
Contents

StepTalk Scripting Introduction - Legend

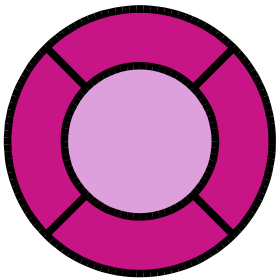
Human



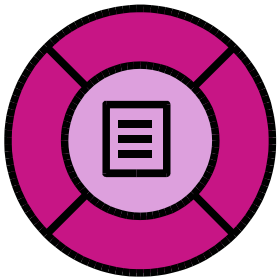
User interface, application, window



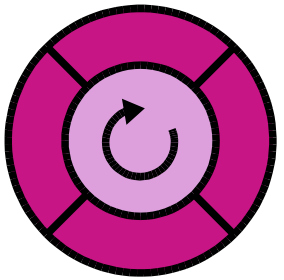
Script



An object



Model object



Controller object

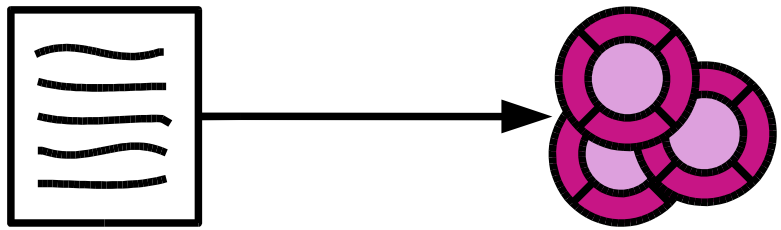
Relationship

Bi-directional relationship

Temporary relationship

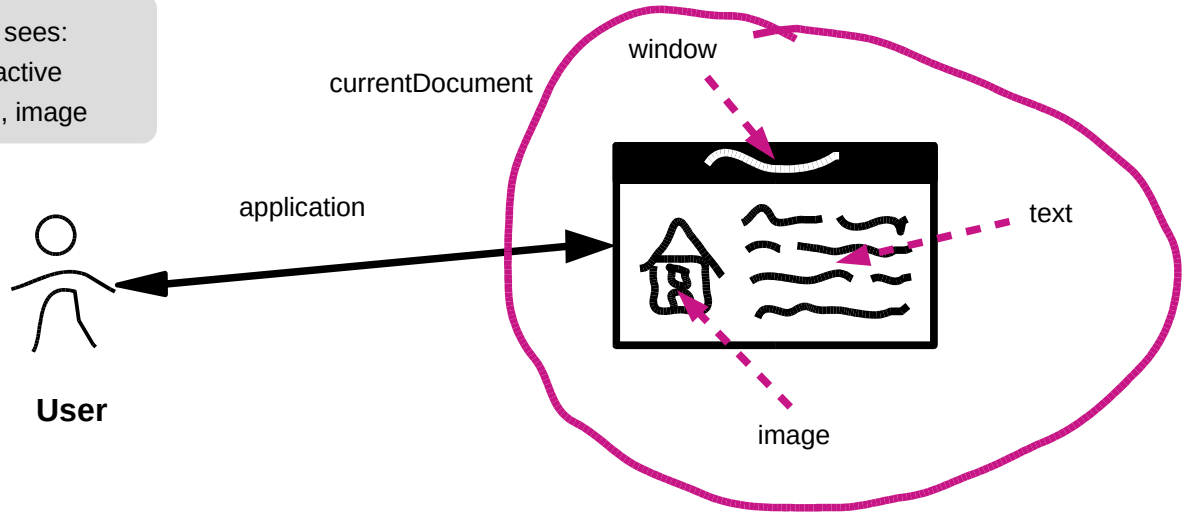
Scripting Session

How to create a scripting session
and how to communicate with objects using scripts

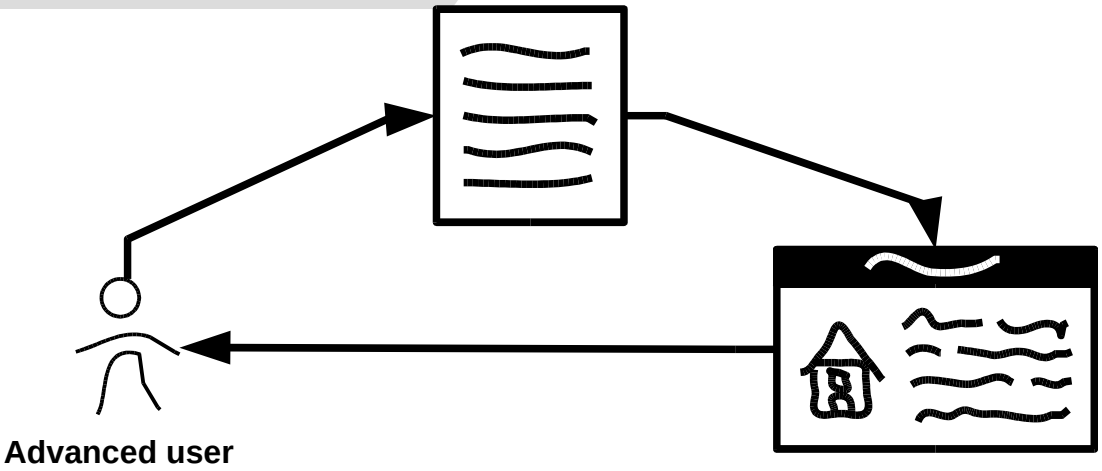


Before StepTalk Scripting

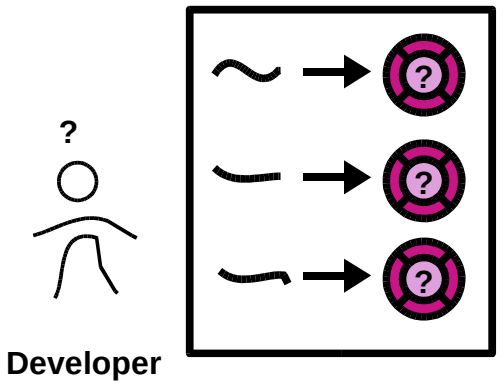
1 **User thinks** of and sees: current document, active window, application, image



2 Advanced user would like to talk to those objects, automate several tasks or create complex tasks that would take lots of time if done manually.

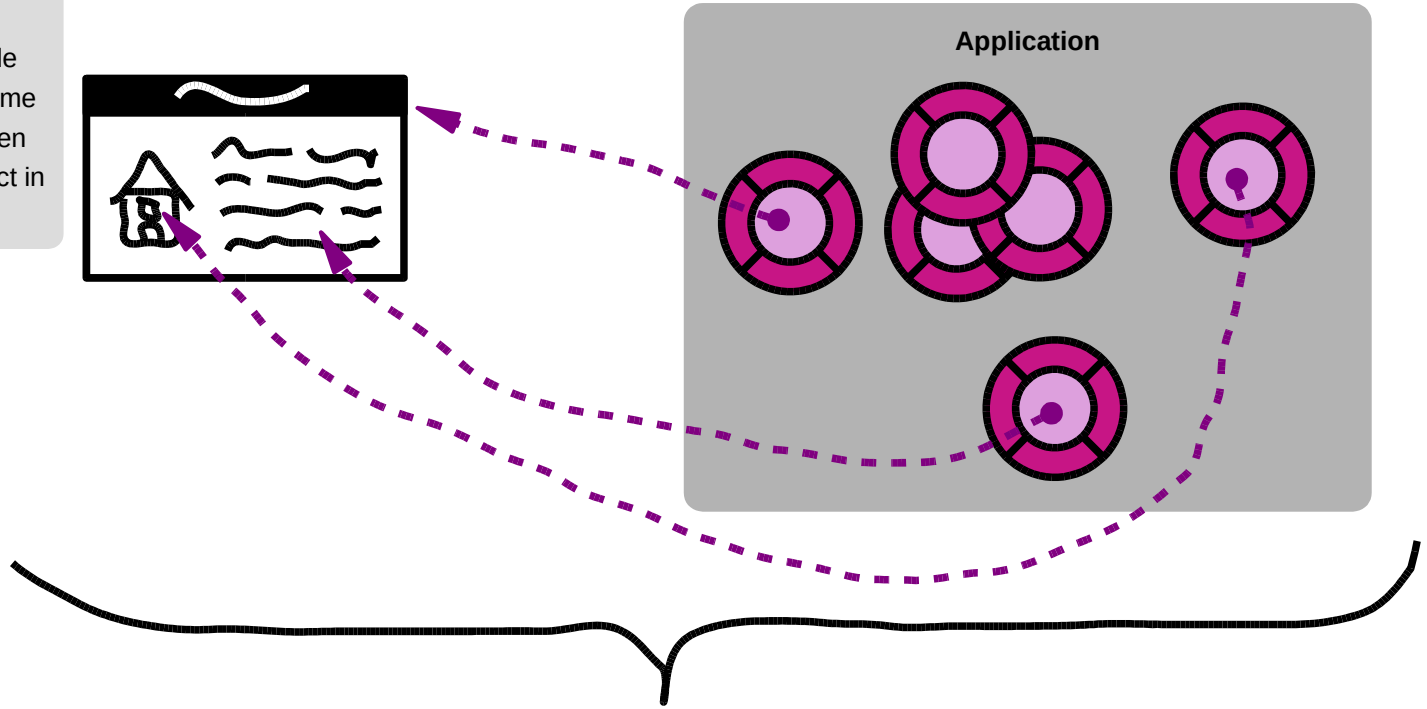


3 ... Therefore the developer has to map entities, that user needs to handle, to real objects inside the application.

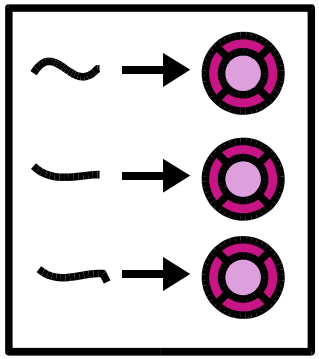


Before StepTalk Scripting

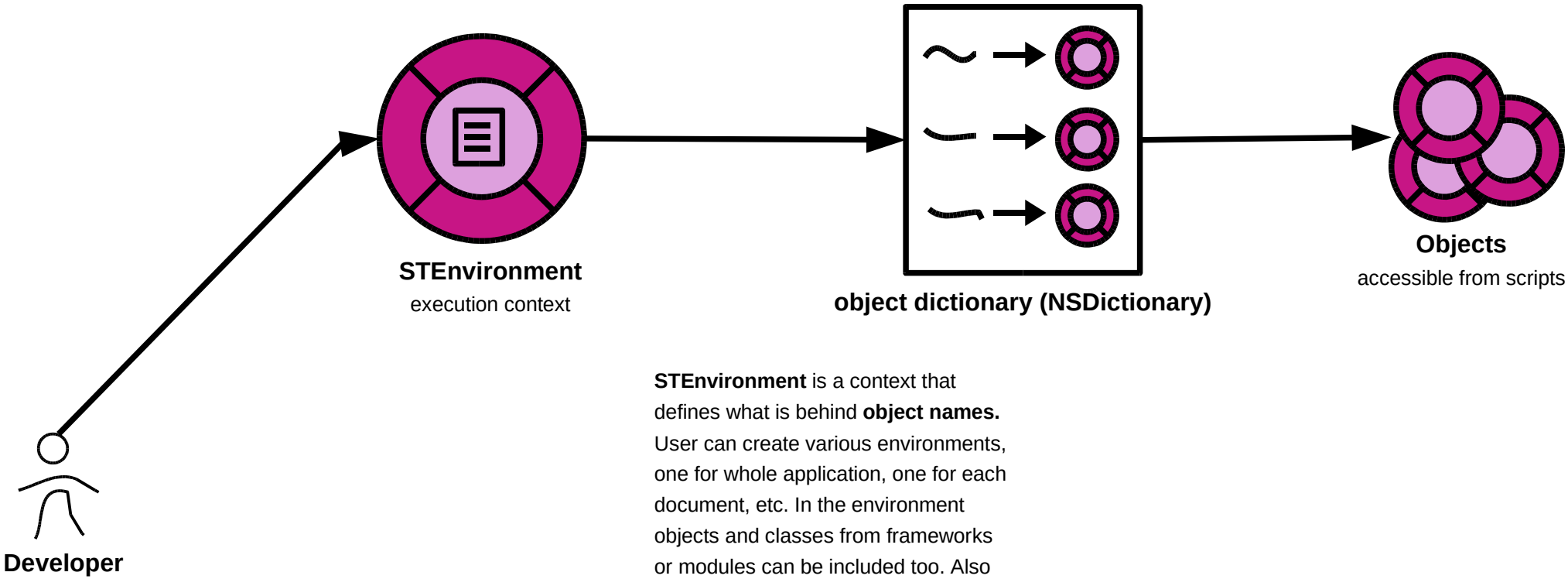
4 Each user-visible entity is somehow represented inside the application. Let us assume that each entity on the screen is a presentation of an object in the application.



5 The developer prepares a mapping between words and objects inside a given context.



StepTalk Scripting

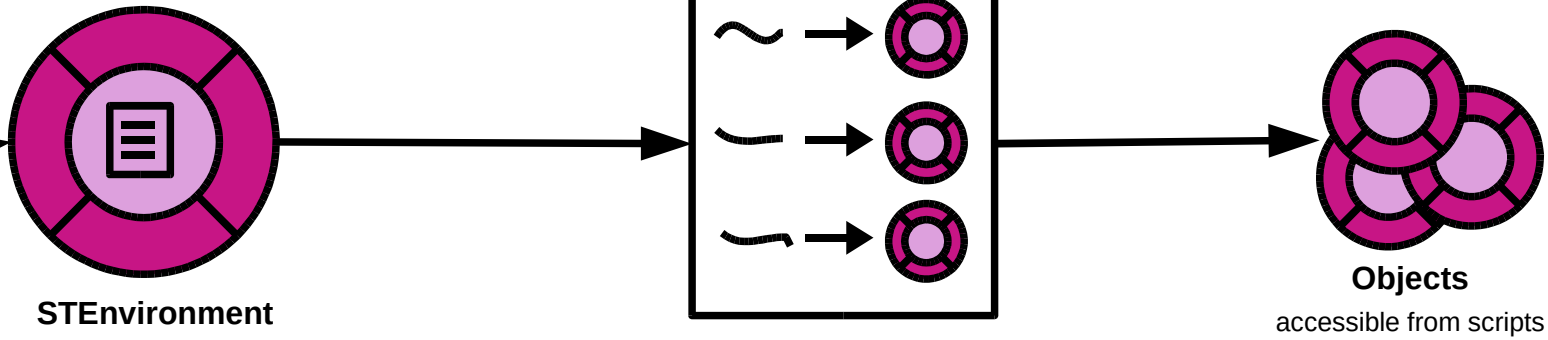


STEnvironment is a context that defines what is behind **object names.** User can create various environments, one for whole application, one for each document, etc. In the environment objects and classes from frameworks or modules can be included too. Also object finders can be associated for finding unknown objects or distant objects.

6

Developer knows objects what are available for scripting. He creates an **environment** (conversation context). This context will let the computer resolve object names. In other words, computer will know, what is meant by 'document', 'application' or 'selectedObject'.

StepTalk Scripting

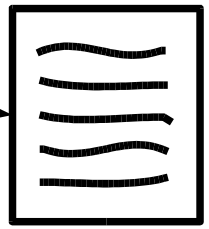


STEnvironment
execution context

object dictionary (NSDictionary)

Objects
accessible from scripts

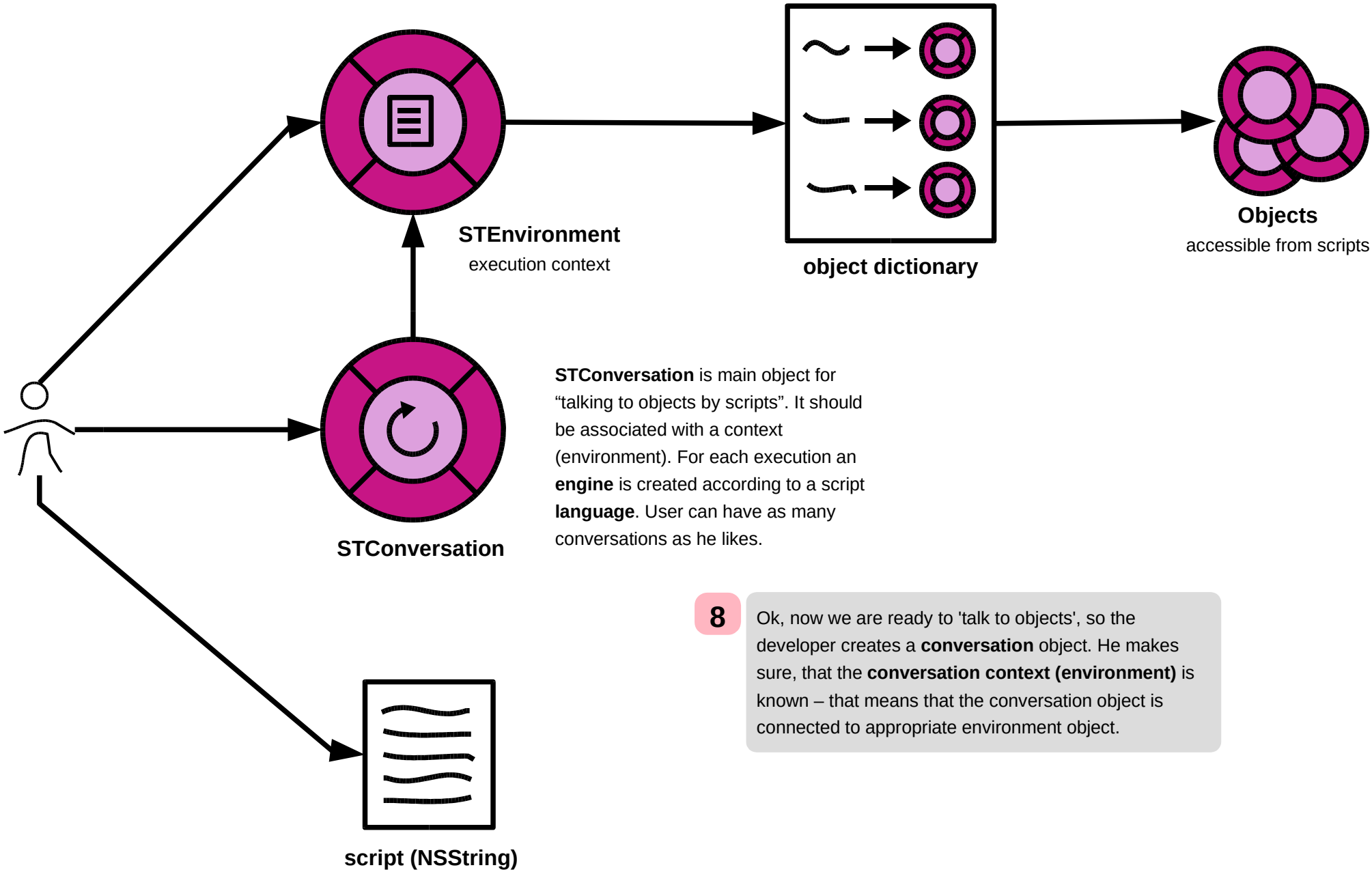
7 Developer loads or creates a **script** – small program in user's preferred programming **language**. User should know in what language the script is written in.



script (NSString)

Script is written in any StepTalk **language**. The user should specify what language the script is written in. Script can reference objects by their name. Objects are then resolved according to the associated **execution context** (environment)

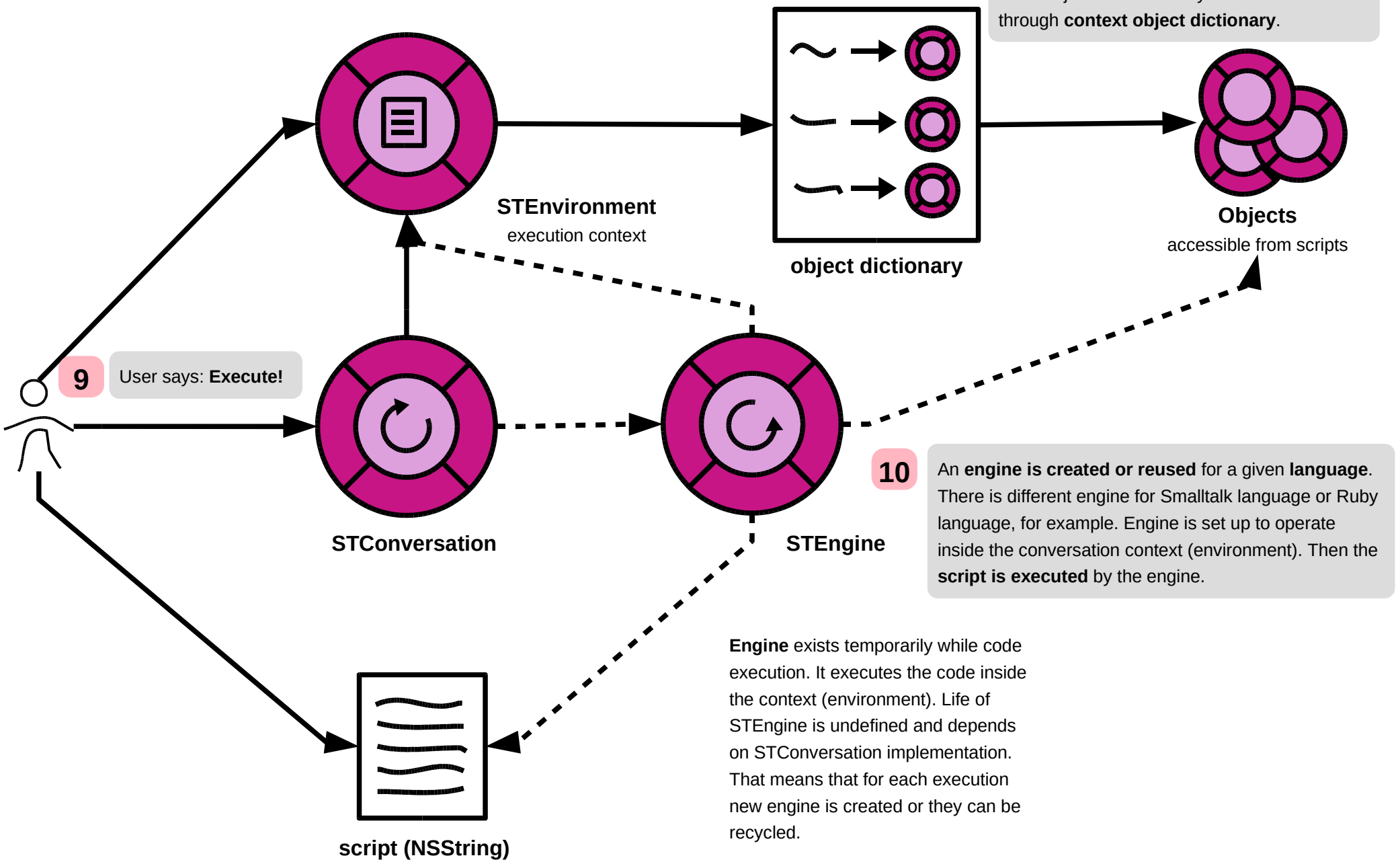
StepTalk Scripting



STConversation is main object for "talking to objects by scripts". It should be associated with a context (environment). For each execution an **engine** is created according to a script **language**. User can have as many conversations as he likes.

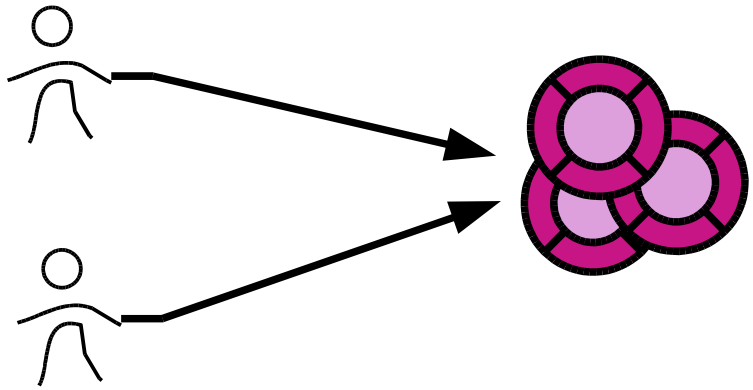
8 Ok, now we are ready to 'talk to objects', so the developer creates a **conversation** object. He makes sure, that the **conversation context (environment)** is known – that means that the conversation object is connected to appropriate environment object.

StepTalk Scripting



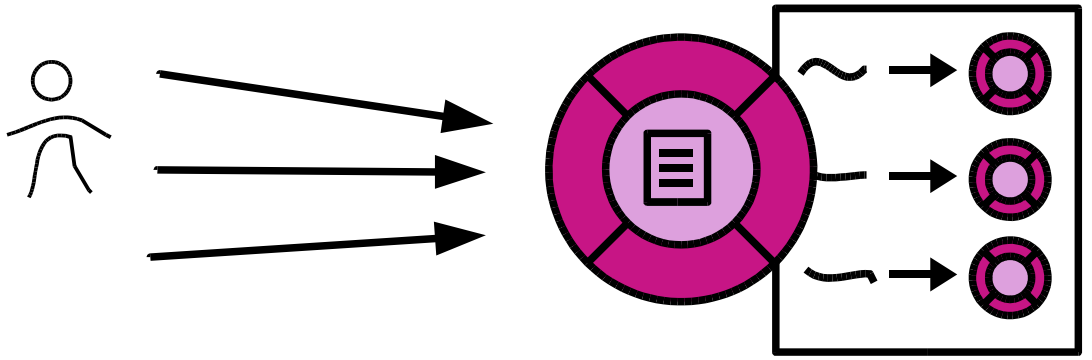
Remote Scripting

Scripting in semi-persistent shared environment



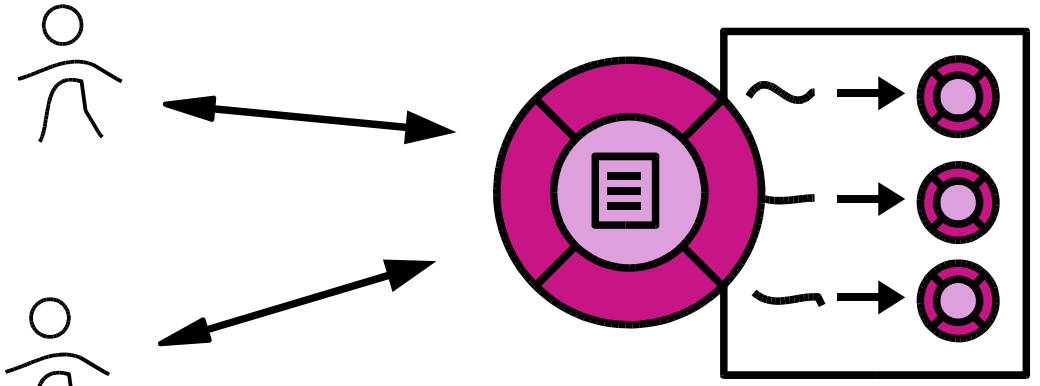
Before Remote Scripting

1a User wants to talk to objects more than once in different times and from different places (processes or applications). He needs the scripting environment with its objects to stay somewhere and to be ready for communication. Therefore he needs **semi-persistent environment** where objects are not destroyed after each conversation and can be reused.



STEnvironment
with objects

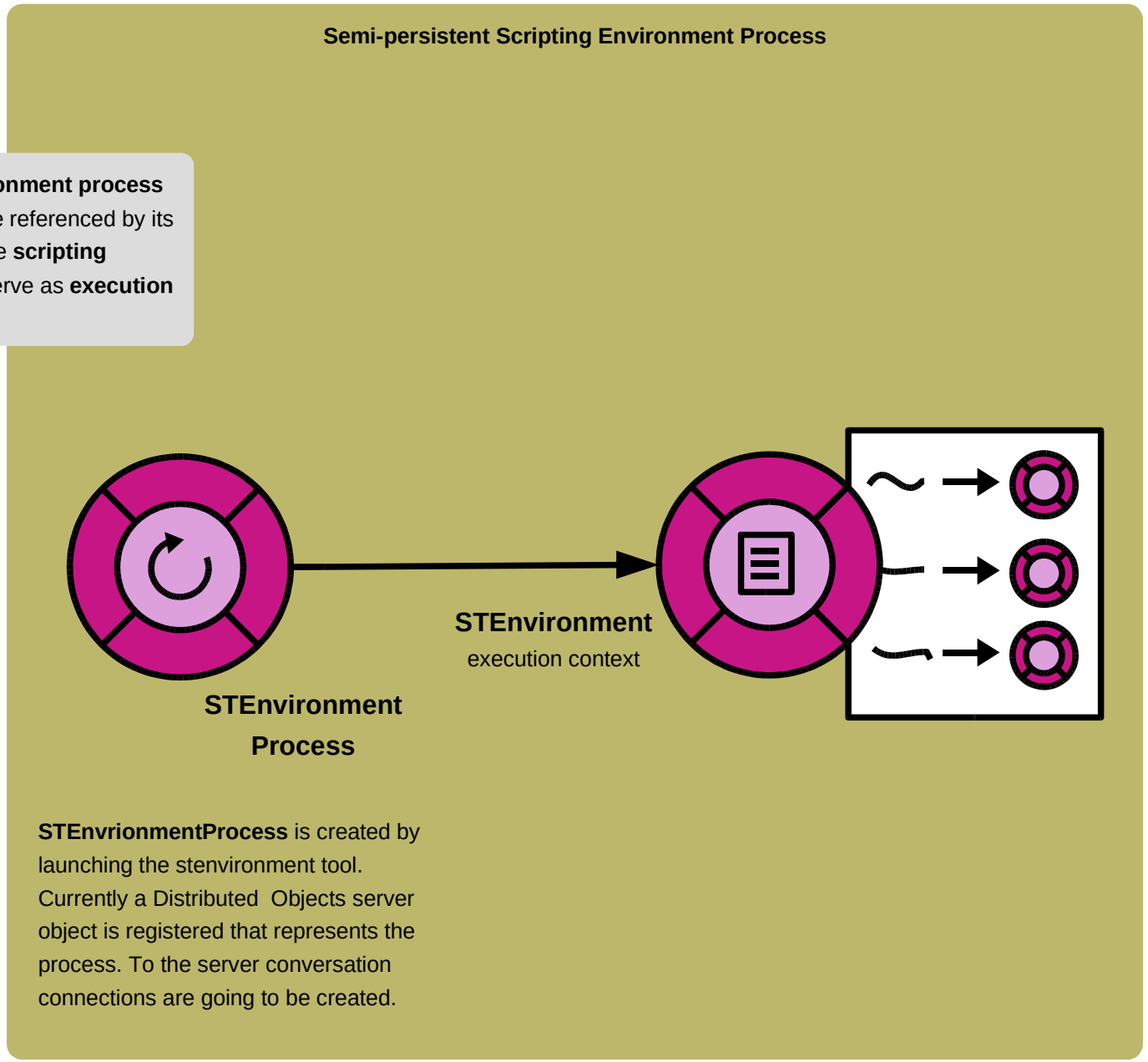
1b Another user wants to bind partial tasks from several applications into a single, larger task. Therefore he needs a **shared environment** for all involved applications.



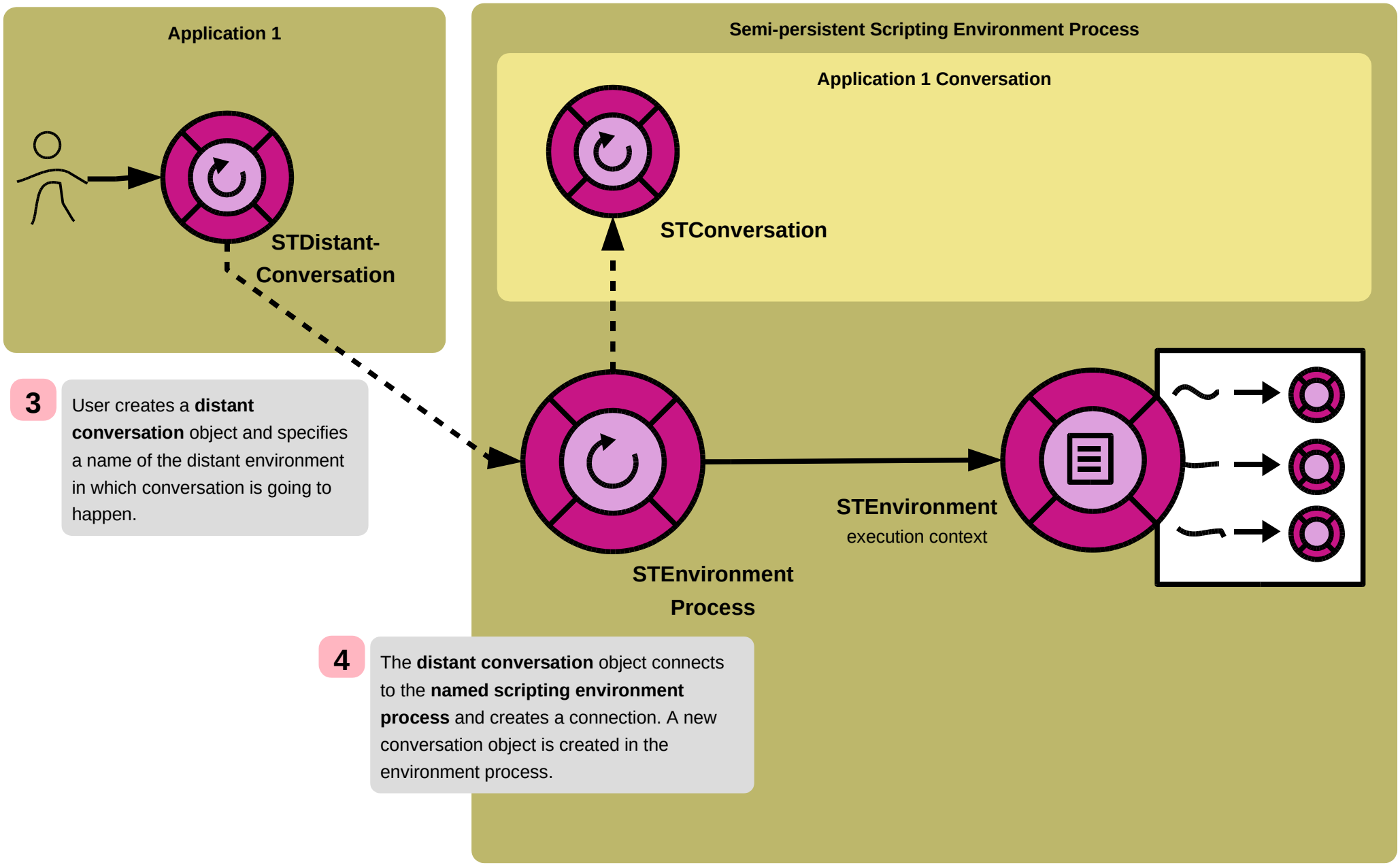
STEnvironment
with objects

Remote Scripting

2 **Semi-persistent Scripting Environment process** is created. The environment will be referenced by its **name**. The process contains single **scripting environment object** that would serve as **execution context** for future scripts.



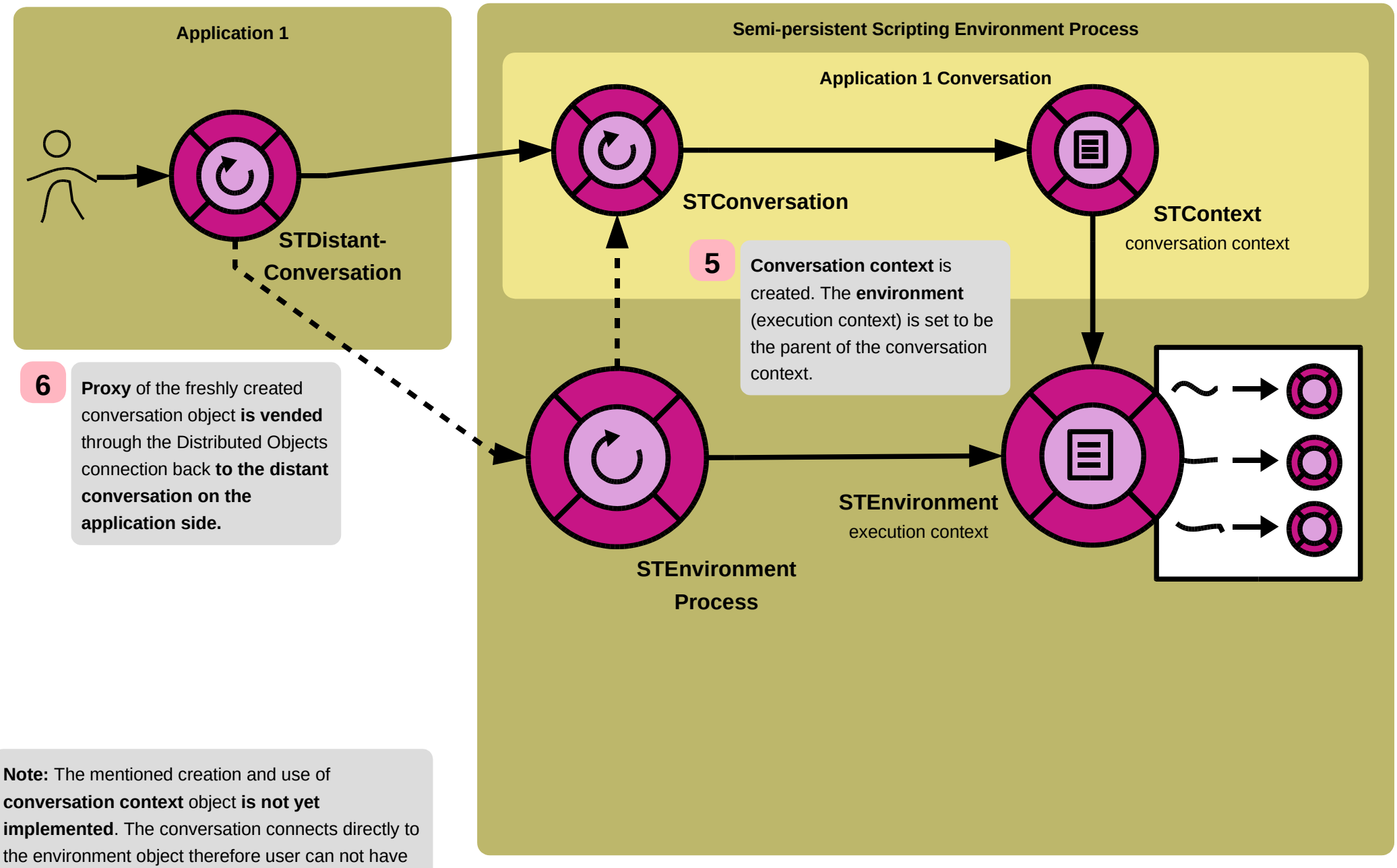
Remote Scripting



3 User creates a **distant conversation** object and specifies a name of the distant environment in which conversation is going to happen.

4 The **distant conversation** object connects to the **named scripting environment process** and creates a connection. A new conversation object is created in the environment process.

Remote Scripting



Note: The mentioned creation and use of **conversation context** object **is not yet implemented**. The conversation connects directly to the environment object therefore user can not have conversation specific objects, **everything is shared**.

Remote Scripting