
Overview

January 18, 2023



Contents

Introduction	2
Architecture	2
Security model	5
Continue with	5

*[CMR]: Context Management Registry

Introduction

Manatee exposes interfaces that allow external applications to interact with its context management and automation features. The primary integration mechanism is through a set of CCOW interfaces. CCOW defines a protocol which orchestrates synchronisation and sharing of state between applications on the same desktop.

The shared state is called the *common context* and changes to it are executed transactionally.

All participants must subject themselves to the state of the common context updating their internal state to match it. Similarly they must update the common context once relevant internal state changes.

The standard also deals with an RPC mechanism where applications may request *actions* to be performed by other participants (or agents) through the context manager.

::: tip Assumed knowledge

Some knowledge of the CCOW standard and the common terms defined here are assumed.

:::

Manatee exposes *ContextManager*, *-Data* and *-Action* interfaces among others to implement the CCOW standard.

- *ContextManager* deals with participants that join the common context and control of transactions that modify the context.
- *ContextData* interface deals with making changes to the common context.
- *ContextAction* allows participants to expose actions they can perform and request other participants to perform actions (RPC style interaction)

Architecture

Besides Manatee as the primary on-desktop context manager implementation there are a few other components in play in a running context management system.

The **Context Management Registry** is a the component that keeps runtime- and configuration information about all the elements of the CCOW dance. First and foremost it is where the context manager can go and figure out which participants are available in its current location. Secondly it will contain information about each *external* participant;

- which part of the common context they are interested in and which part of it they are allowed to manipulate
- which actions are exposed by the participant
- how the application representing the participant is configured, how is it launched etc

For *internal* participants it will contain the scripting necessary to extract information from and manipulate the application in order to allow the application to respond to changes in the common context and report its own internal state changes back to the context manager.

::: tip External and internal participants

An *external* participant is an application that interacts with Manatee through the APIs that Manatee exposes. An *internal* participant is an application that we automate and through automation enables the application to become a context participant.

:::

Information in the CMR is either configured using the Cuesta web-interface or provided via the API exposed by Kwanza (the CMR implementation). The figure below illustrates the overall architecture.

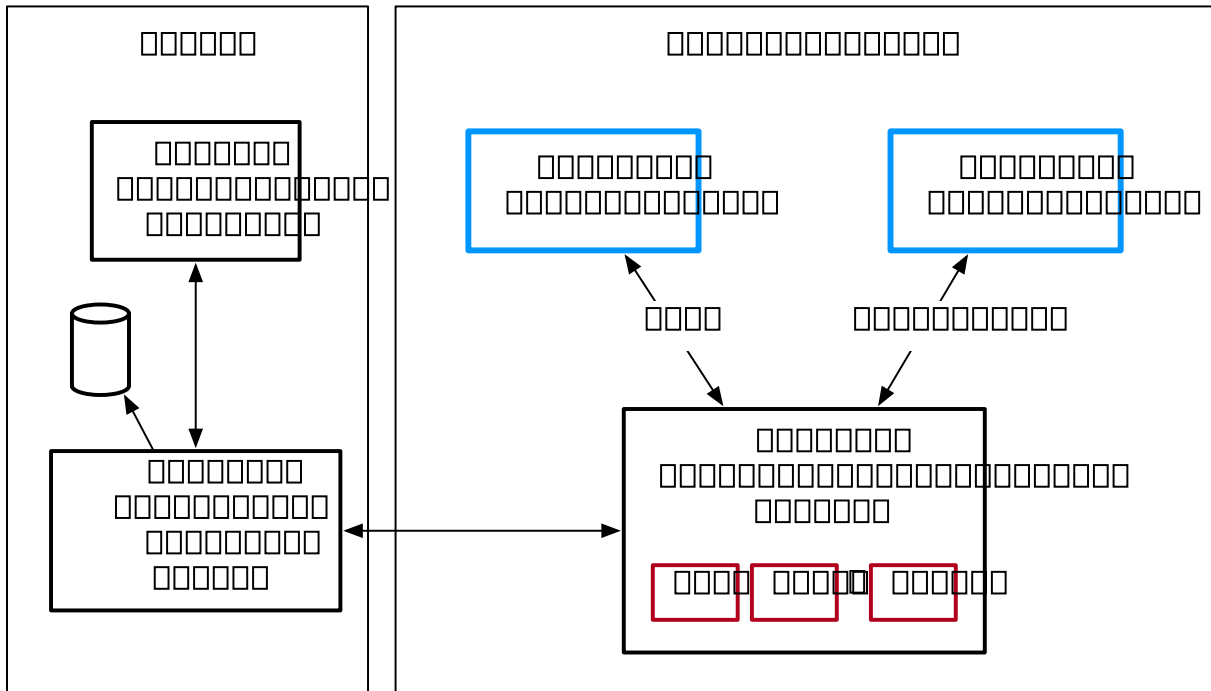


Figure 1: Architecture diagram

There is also the possibility of running a centralised context manager and then having Manatees report into that instance (seen below). This allows context synchronisation across multiple devices or in a Citrix-style environment.

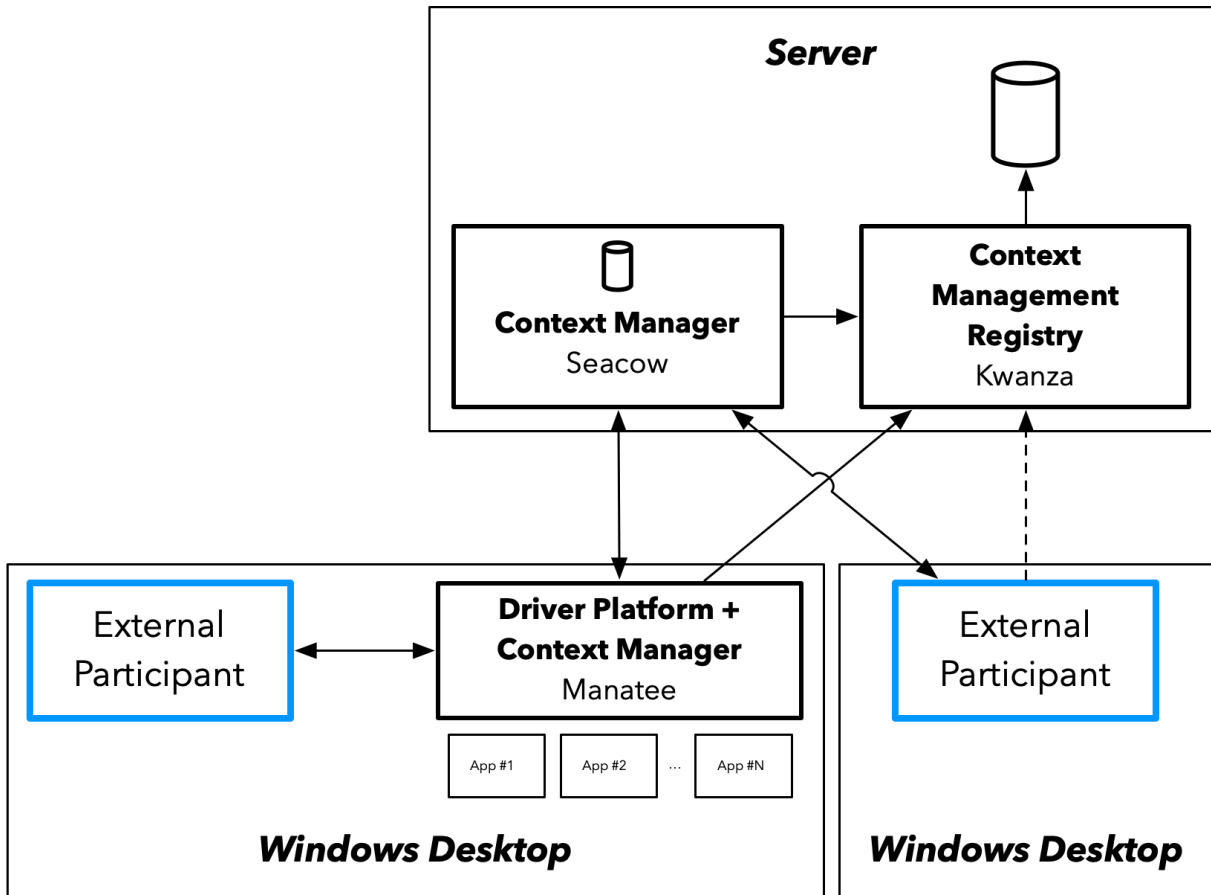


Figure 2: Remote CM diagram

Finally Manatees can report to each other directly if setting up a centralised CM is not an option.

Security model

All communication channels offer encrypted options. Authentication is needed to configure the system and thus to get access to the keys needed for identification and connection to the context manager.

Continue with ...

Read about the protocols that you can use to connect to Manatee with:

- JSON-RPC
- GRPC