ALPACA INTRODUCTION

Concepts and Use

Abstract

In 2000, ASCOM brought to astronomy software on the Windows platform a standard way to communicate between programs and devices such as mounts, cameras, etc.

Alpaca extends the ASCOM standard interfaces into the realm of distributed systems on multiple OS platforms while maintaining its language independence. This is achieved by use of Representational State Transfer (REST), JSON, HTTP and TCP/IP technologies.

For the first time, this removes the dependency that that ASCOM clients and drivers can only be developed and run on the Windows operating system.

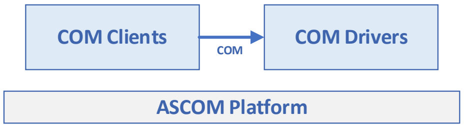
Peter Simpson, Bob Denny
peter@peterandjill.co.uk, rdenny@dc3.com

Contents

1. Introduction

ASCOM interfaces define a set of standard operations for classes of astronomy devices such as mounts and focusers etc. These operations are independent of any vendor and technology. To be useful though, these interfaces need to be brought to life in a technology set that both client applications and astronomy devices can use.

Historically, there was just one implementation of the interfaces, in Microsoft's Component Object Model (COM) technology, which only runs on the Windows OS.



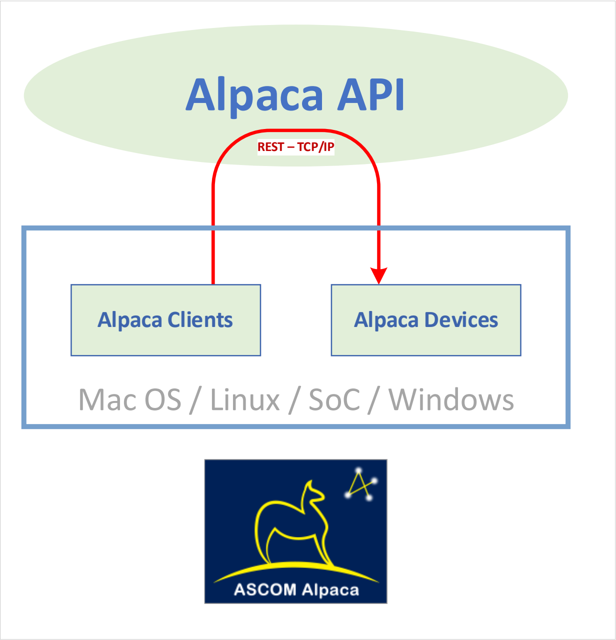
2. The ASCOM Alpaca API

Alpaca is a second implementation of the interfaces:

- Alpaca uses **TCP/IP, HTTP, REST and JSON** technologies.
- Alpaca can be used on **any operating system or device** that supports networking, and the technologies listed above.
- Alpaca has **no dependency on Windows** of any kind.

Since Alpaca employs widely used network protocols, Unix based operating systems such as Linux and MacOS can host ASCOM Alpaca clients together with multiple Alpaca devices on a single or distributed across multiple devices.

Furthermore, the networked nature of Alpaca opens possibilities of using Raspberry PI / Arduino / SoC hardware as Alpaca devices.



The Alpaca APIs are functionally identical to the COM-based ASCOM APIs while adding cross-platform and distributed operation.

2.1 Alpaca API Documentation

The ASCOM Alpaca APIs are documented using the Swagger OpenAPI toolset and are available through a URL on the ASCOM Standards web site. The ASCOM API is fully documented here:

https://www.ascom-standards.org/api

To start exploring go to the above API URL and click a grey Show/Hide link to expand one of the sets of methods and then click the blue GET or orange PUT methods for detailed information on that API call. Note that this documentation, along with the companion ASCOM Remote Server Management API is documented in the next section.

Anyone who is familiar with the ASCOM COM based APIs will immediately feel at home with the functionality available through ASCOM Alpaca. Full details on how to construct, send and decode ASCOM Alpaca API calls are given in the companion Alpaca API Reference document[1].

2.2 Discovery

ASCOM clients need to find ASCOM devices in order to use them. The COM implementation effects this through the ASCOM Profile and COM's own well known object name (ProgID) mechanic.

Alpaca clients use a networked discovery mechanic to locate Alpaca devices on the client device's broadcast domain (usually the local LAN segment, but network administrators can configure this to be a larger scope if required). Further information on the Discovery mechanic is available in the ASCOM Alpaca API Reference[1].

3. Interoperability, Transition and Co-existence

Today, ASCOM is a rich eco-system of Windows clients and drivers that communicate through COM rather than Alpaca. Without bridging technologies, these clients and drivers will be an isolated technical continent cannot be accessed by Alpaca clients and that is unable to communicate with Alpaca devices.

To address this, two bridging technologies have been developed:

- **ASCOM Remote Server** - Enables Alpaca clients to discover and use COM drivers
- **ASCOM Dynamic Clients** - Enable COM clients to discover and use Alpaca devices

3.1 ASCOM Remote Server

The ASCOM Remote Server is a Windows program that runs on the PC that hosts the COM drivers.

The Remote Server provides a presentation layer that translates incoming networked Alpaca protocol instructions from Alpaca clients into COM calls that are passed to the configured COM drivers. The Remote Server then translates COM driver responses into Alpaca responses that are returned to the Alpaca clients.

3.2 ASCOM Dynamic Clients

ASCOM Platform 6.5 introduced Alpaca Dynamic Clients, which COM clients see as traditional COM drivers. The Dynamic Clients translate COM client instructions into the Alpaca protocol, send them to

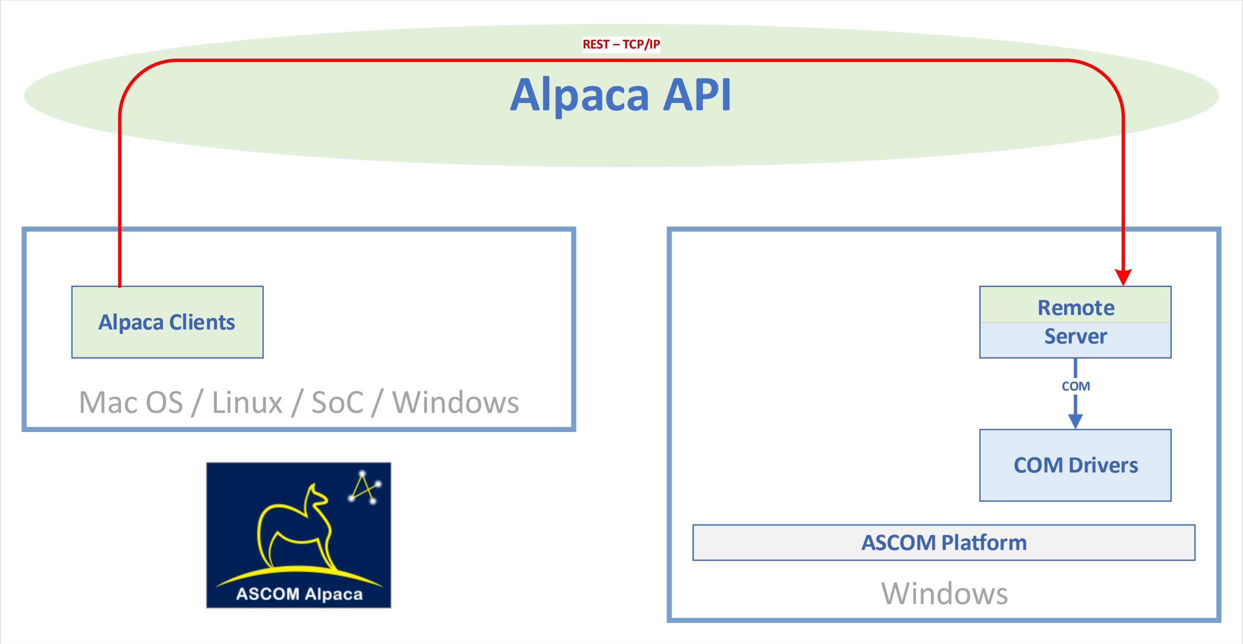
[1] API Reference Document LINK

a remote Alpaca device and translate the device's responses back to a form that makes sense to the COM client.

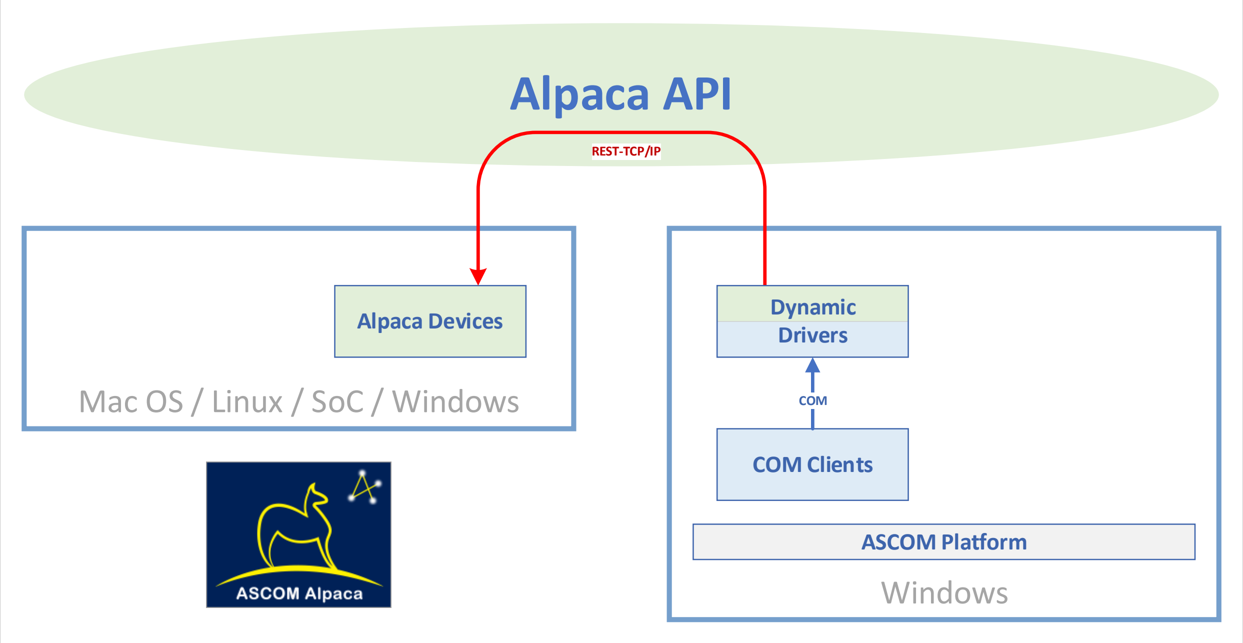
3.3 Interoperability Configurations

These diagrams show potential operating configurations enabling communication between ASCOM Alpaca and COM devices:

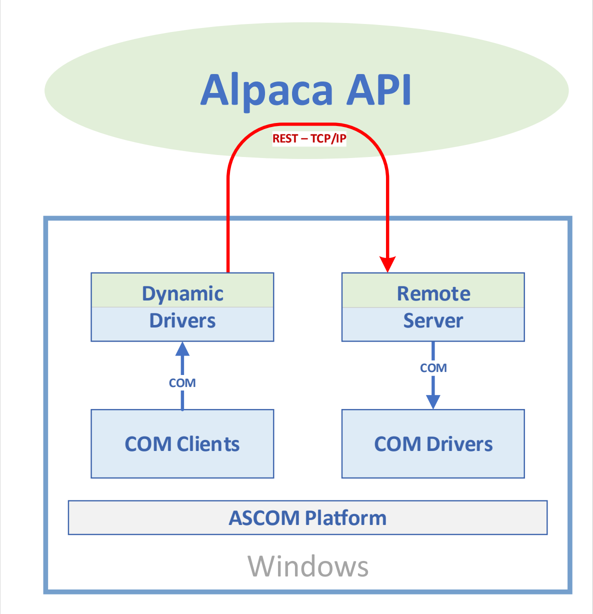
3.3.1 Alpaca clients using COM drivers hosted on a Windows PC



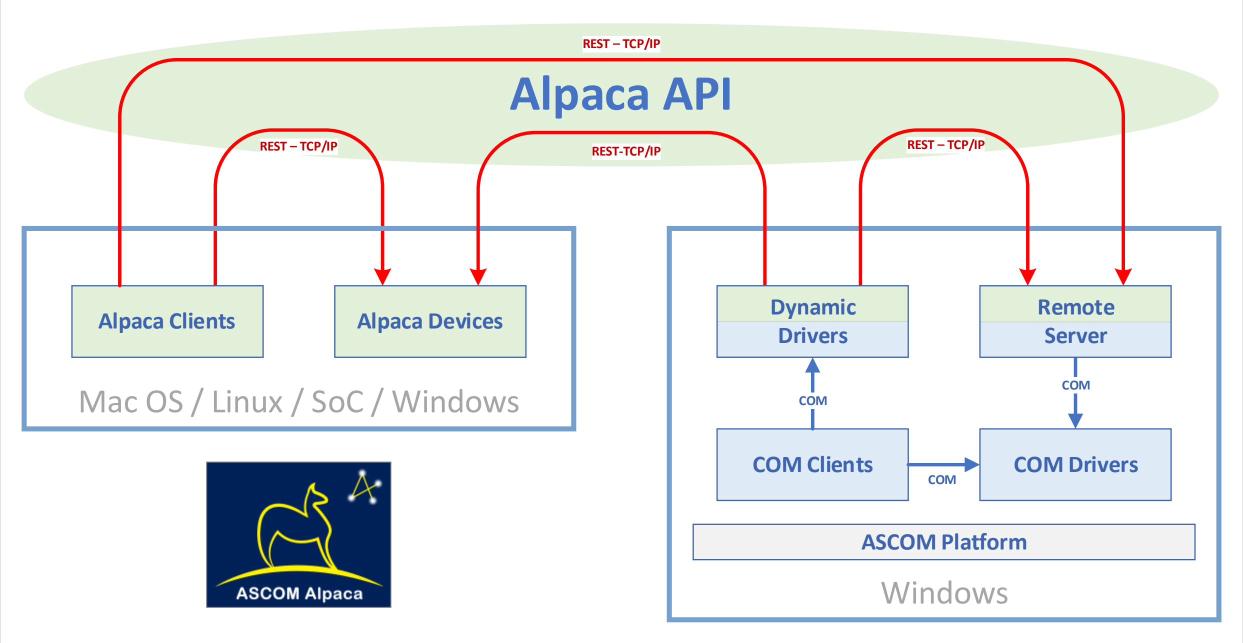
3.3.2 Windows COM clients using Alpaca devices running on Linux / MacOS / SoC



3.3.3 Windows COM clients using COM drivers running on a different PC



3.3.4 Summary of all ASCOM operating and interoperability options


REST – TCP/IP
Alpaca API
REST – TCP/IP
REST-TCP/IP
REST – TCP/IP
Alpaca Clients
Alpaca Devices
Mac OS / Linux / SoC / Windows
ASCOM Alpaca
Dynamic Drivers
Remote Server
COM
COM
COM Clients
COM
COM Drivers
ASCOM Platform
Windows