3.2.2 API formats and protocols

APIs operate based on requests and responses. A server makes a request of the API, and the API response. This interaction is made possible by formats and protocols. Formats specify how data can be sent to and accessed by other applications, while protocols are the rules that define how applications interact with each other. The most popular web protocol is HTTP, which you will learn about it later in this unit. It is not the only one though. Other formats or protocols APIs use include SOAP, REST, XML-RPC and JSON-RPC. You will learn about some of these in more detail as the unit progresses. For now, let's compare SOAP and REST as sets of formats and protocols, and then briefly consider XML-RPC and JSON-RPC.

SOAP vs REST

SOAP (which stood for Simple Object Access Protocol when it was created, but is no longer its name) is a standards-based web access protocol, developed in the late 1990s. SOAP's aim was to help define messages used by applications and exchanged between systems, allowing them to operate in most enterprise environments. SOAP uses XML (Extensible Markup Language) to provide messaging services, and is highly extensible, meaning it is designed to support expansion. SOAP functions as an envelope, encoding the requests and responses of the API.

XML

XML is a standard way to encode documents and data. It can be used on any platform, and does not contain default tags. This means that in each file, tags need to be defined.

Use this resource to learn more about XML:

- **XML Introduction (https://www.w3schools.com/xml/xml_whatis.asp)**

REST, or REpresentational State Transfer, is an architectural style introduced in 1999. Unlike SOAP, REST relies on HTTP specifications to operate and make use of the URLs to make data available through the web. REST also works well with mobile applications—indeed it was only after 2008 and the growth of iPhone usage that REST really took off. REST is lightweight, simple, and widely recognised and many teams use it as a starting point in developing APIs. REST also functions as a way of conveying the requests and responses of

the API. In contrast to SOAP as an envelope, you can think of REST as a postcard: a simpler, lightweight, but less secure, way of transmitting information.

Comparison of SOAP and REST

	SOAP	REST
Style	Protocol	Architectural
Function	Function-driven; transfer-structured information	Data-driven; access a resource for data
Data format	XML only	Allows many data formats, including plain text, HTML, XML, and JSON
Security	WS (Web Services) Security and SSL	SSL and HTTPS
Bandwidth	High	Low
Data cache	Cannot be cached	Can be cached
Payload handling	Strict communication contract	Needs no knowledge of the API

COS60016: SOAP vs REST (2021) created by Swinburne Online.

Remote Procedure Calls

RPC, or Remote Procedure Call, is the earliest form of API. Procedures, or functions, have been the dominant means of constructing computer code since they were introduced in 1958. As you have noticed so far in this unit, functions are a central organising construct of Python. The ubiquity of procedures in codes meant that procedures became the primary way of designing and implementing APIs through, you guessed it, RPCs.

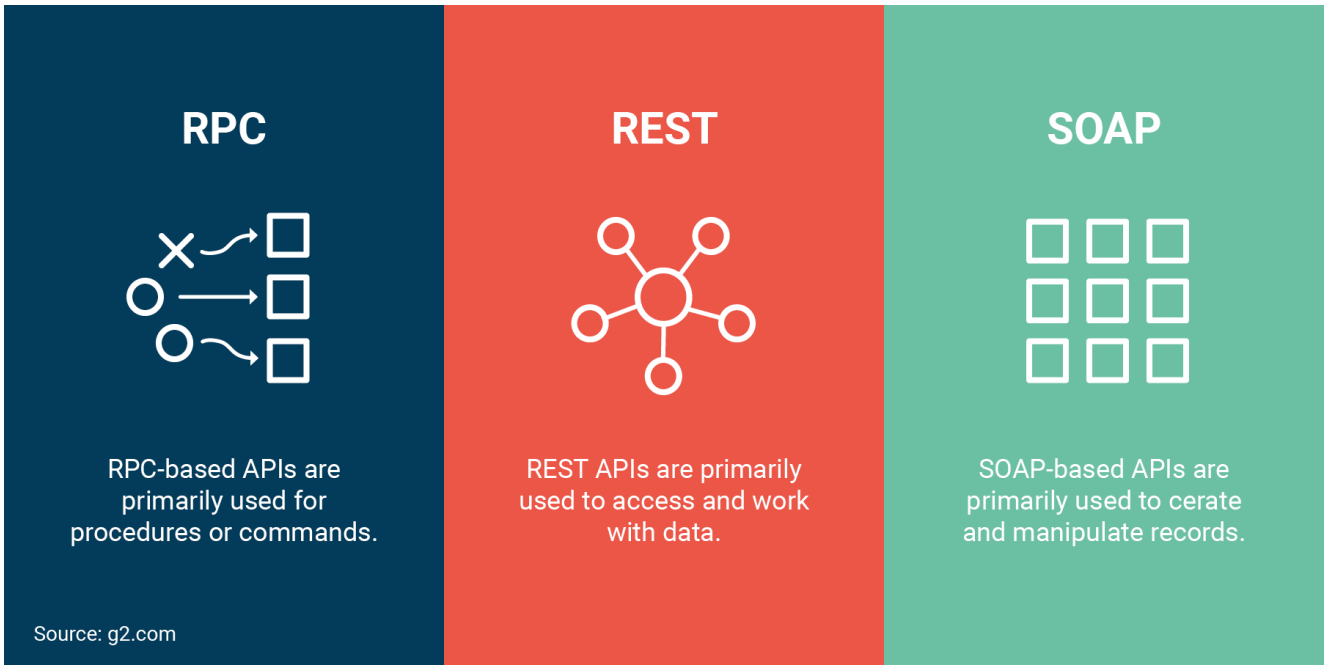
Essentially, an RPC works in the same way as calling a function in any other language, be it Python, Java, C++, PHP, and so on. This means RPCs are a great fit for the thought processes and skills of programmers on both the production and consumption side of APIs. This is because calling a remote procedure or learning the procedures of a remote API is similar to calling a regular programming language procedure (Nally 2018). As a result, RPC APIs are much quicker and easier to implement than SOAP, for example.

XML-RPC and JSON-RPC

XML-RPC is an RPC that uses Extensible Markup Language (XML) to format data and HTTP as the means of transport. XML-RPC was the basis for SOAP. You can read more about it in the resource **What is XML-RPC? (https://tldp.org/HOWTO/XML-RPC-HOWTO/xmlrpc-howto-intro.html)** (tldp n.d.).

JSON-RPC is an RPC that provides a simple format to utilise HTTP. Initially developed to make use of JavaScript messaging, JSON-RCPs has grown to be supported by all languages, and is easy to adopt, format, debug, and understand. Indeed, it has become the format of choice for most developers, and you will learn more about it later in the week.

Each API format suits different tasks. The following image summarises where SOAP, REST, and RPC APIs work best.



COS60016: RPC, REST, and SOAP (2021) adapted from Das.

Reflection

At this point, you may not have an opinion about which protocol you'd want to use, but it is something that some programmers feel very strongly about. Search on **Stack Overflow (http://stackoverflow.com)** for 'SOAP vs REST' or 'REST vs RPC' (Stack

Overflow n.d.). Why do you think REST has become so popular compared to alternatives?