TECHNICAL

What is SOAP? REST? What are the differences between the two?

In general, REST is more lightweight and is easier to scale.

SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) are both web service communication protocols. SOAP was long the standard approach to web service interfaces, although it's been dominated by REST in recent years. Both are ways to access and interact with data, but REST has the advantage of being simpler to use and supporting different data formats, meaning we can now support simpler and more flexible integrations with systems or customers looking to integrate Web of Science data into their workflows.

	SOAP (*older*)	**REST (*newer*)**
Meaning	Simple Object Access Protocol	Representational State Transfer
Clarivate Portal Implications	Non-portal-enabled	Portal-enabled
Design	Standardized protocol with pre-defined rules to follow.	Architectural style with loose guidelines and recommendations.
Approach	Function-driven (data available as services, e.g.: "getUser")	Data-driven (data available as resources, e.g. "user").
Caching	API calls cannot be cached.	API calls can be cached.
Security	WS-Security with SSL support. Built-in ACID compliance.	Supports HTTPS and SSL.
Performance	Requires more bandwidth and computing power.	Requires fewer resources.
Message format	Only XML.	Plain text, HTML, XML, JSON, YAML, and others.
Transfer protocol(s)	HTTP, SMTP, UDP, and others.	Only HTTP
Recommended for	Enterprise apps, high-security apps, distributed environment, financial services, payment gateways, telecommunication services.	Public APIs for web services, mobile services, social networks.
Advantages	High security, standardized, extensibility.	Scalability, better performance, browser-friendliness, flexibility.
Disadvantages	Poorer performance, more complexity, less flexibility.	Not suited to build complex secure schemas (right and roles) – less secure

For an alternate technical breakdown, see here: **https://stackify.com/soap-vs-rest/.**