SOA leads to an architectural style that is an evolution of the RPC (Remote Procedure Call) style, by declaring and invoking operations in a standard and lan- guage independent way

* Resources are peers, in the sense that each resource can both offer a service and invoke other resource’s service
* To use a resource, its service description (i.e., WSDL document) must be obtained first, so that a contract can be established between that resource (the provider) and the resource that uses it (the consumer). That contract is static (design time)
* A concrete schema, with the names used, must be specified and be compatible on both sides, oth- erwise a representation returned by a resource, for example, will not be able to be analyzed and understood.
* Resources send requests to each other to invoke a given operation, be it with a SOA or REST approach. These requests and their responses usually contain data, which are serialized, sent and reconstructed upon reception of the corresponding message (Figure 2). The sender and receiver need to interpret those data in a com- patible way, which means interoperability at the schema level between the corre- sponding data structures at both ends.
* interoperability is simply based on sharing schema files, typically in the world of Web Services, or predefined media types, as used by RESTful applications. In either case, data types need to be fully known by both interacting resources.
* we show how interaction is still possible with only a partial knowledge of types, as long as the characteristics actually used are included (partial interoperability). This is a way of getting closer to solving the fundamental integration problem, by reducing coupling to what is actually required.