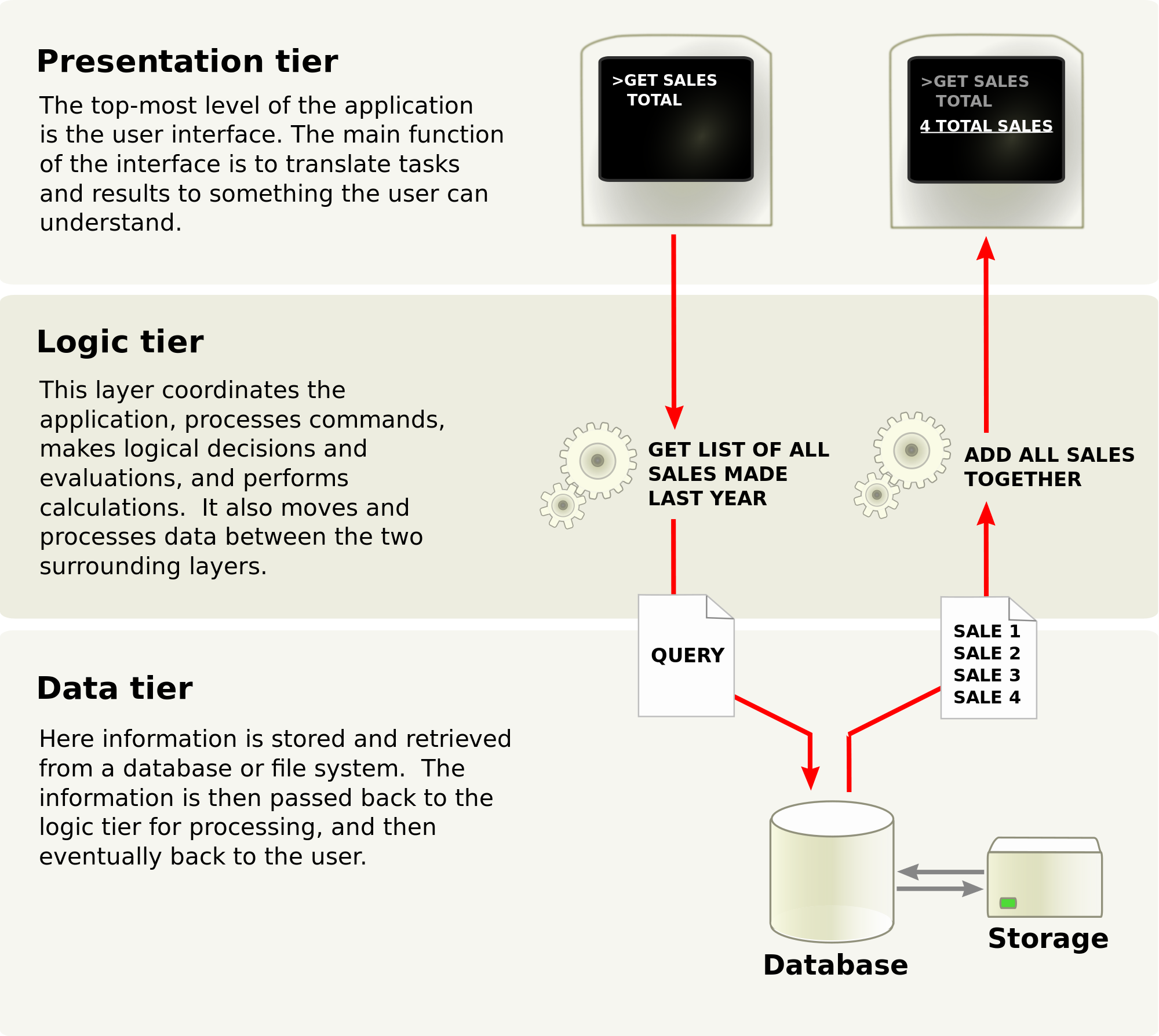
**How an application is made up?**

An application is made of three essential parts that ideally should be able to work and communicate in a segmented way, so one could be swapped out for another:

1. Data Tier: Where data is retrieved from the database and file system and then stored.
2. **Logic Tier:** The brain of the application, this processes the data between the layers, coordinating the application, processing commands, and making logical decisions. **This layer is made of the API.**
3. Presentation Tier: This top layer of the app is the user interface, which translates tasks into something the user understands.



In other words the API is the brain of our connected world. It is the set of tools, protocols, standards and code that glues our digital world together. APIs allow for companies to become more agile, for things to go mobile, and everything to work together in a streamlined, integrated way.

[**What is the difference between API and web services?**](http://stackoverflow.com/questions/4229567/what-is-the-difference-between-api-and-web-services)

Definitions from Wikipedia:

API:  
An application programming interface (API) is a set of routines, data structures, object classes and/or protocols provided by libraries and/or operating system services in order to support the building of applications.

Web service :  
A Web Service is defined by the W3C as "a software system designed to support interoperable machine-to-machine interaction over a network".

Clearly, both are means of communications. The diference is that Web Service almost always involves communication over network and HTTP is the most commonly used protocol. Web service also uses SOAP, REST, and XML-RPC as a means of communication. While an API can use any means of communication e.g. DLL files in C/C++, Jar files/ RMI in java, Interrupts in Linux kernel API etc.   
So, you can say that-  
  
1. Web Service is an API wrapped in HTTP.  
2. All Web Services are API but APIs are not Web Services.  
3. Web Service might not perform all the operations that an API would perform.  
4. A Web Service needs a network while an API doesn't need a network for its operation.

**Why we want to test the API?**

The API is what gives the value to the application. If an API breaks because errors weren’t detected, it could not only break a single application but a chain of business processes hinged to it.

**How to access Web services?**

Simple Object Access Protocol (SOAP) and Representational State Transfer (REST) are two answers for this question.

**What is SOAP?**

SOAP is XML based protocol. It is platform independent and language independent. By using SOAP, you will be able to interact with other programming language applications.

**What are the advantages** **of SOAP?**

* **WS Security**: SOAP defines its own security known as WS Security.
* **Language and Platform independent**: SOAP web services can be written in any programming language and executed in any platform.

**What are the disadvantage of SOAP**?

**Slow**: SOAP uses XML format that must be parsed to be read. It defines many standards that must be followed while developing the SOAP applications. So it is slow and consumes more bandwidth and resource.

**WSDL dependent**: SOAP uses WSDL and doesn't have any other mechanism to discover the service.

**What is REST?**

RESTful Web Services are REST architecture based web services. In REST Architecture everything is a resource. RESTful web services are light weight, highly scalable and maintainable and are very commonly used to create APIs for web based applications.

**What are HTTP Methods for RESTful Services?**

The primary or most-commonly-used HTTP verbs (or methods, as they are properly called) are POST, GET, PUT and DELETE.

**GET**

The GET method is used to retrieve information from the given server using a given URI. Requests using GET should only retrieve data and should have no other effect on the data.

**POST**

A POST request is used to send data to the server, for example, customer information, file upload, etc. using HTML forms.

**PUT/update**

Replaces all current representations of the target resource with the uploaded content.

**DELETE**

Removes all current representations of the target resource given by a URI.

**What are the difference between SOAP and REST API?**

|  |  |  |
| --- | --- | --- |
| **No** | **SOAP** | **REST** |
| **1.** | SOAP is a protocol. | REST is an architectural style. |
| **2.** | SOAP stands for Simple Object Access Protocol. | RESTstands for Representational State Transfer. |
| **3.** | SOAP can't use REST because it is a protocol. | REST can use SOAP web services because it is a concept and can use any protocol like HTTP, SOAP. |
| **4.** | SOAP uses services interfaces to expose the business logic. | REST uses URI to expose business logic. |
| **5.** | JAX-WS is the java API for SOAP web services. | JAX-RS is the java API for RESTful web services. |
| **6.** | SOAP defines standards to be strictly followed. | REST does not define too much standards like SOAP. |
| **7.** | SOAP requires more bandwidth and resource than REST. | REST requires less bandwidth and resource than SOAP. |
| **8.** | SOAP permits XML data format only. | REST permits different data format such as Plain text, HTML, XML, JSON etc. |
| **9.** | SOAP is less preferred than REST. | REST more preferred than SOAP. |

**What are the elements of the REST Messages?**

**Endpoint:**The endpoint is the URL of the data to be retrieved, updated, or deleted. Every REST message must specify an endpoint.

**Headers:** HTTP headers in REST messages contain information about the request, such as the desired response format. A REST message may specify any number of headers.

**Authentication setting:**Authentication settings include which type of authentication to use, such as basic auth or OAuth, as well as the credentials to use.

**HTTP methods:**

HTTP methods, such as GET, POST, or DELETE interact with the data at the endpoint.

You can optionally override the parent REST message configuration in each HTTP method such as by specifying a different endpoint, authentication credentials, or headers.

HTTP methods that send content, such as POST, include a message body detailing this content.

A REST message may specify multiple HTTP methods. When sending a REST message, such as through a workflow activity or script, you must specify which HTTP method to use.

**What is JSON Format?**

JSON: JavaScript Object Notation.

JSON is syntax for storing and exchanging data.

JSON is an easier-to-use alternative to XML.

Example:

{"employees":[  
    {"firstName":"John", "lastName":"Doe"},  
    {"firstName":"Anna", "lastName":"Smith"},  
    {"firstName":"Peter", "lastName":"Jones"}  
]}

XML example:

<employees>  
    <employee>  
        <firstName>John</firstName> <lastName>Doe</lastName>  
    </employee>  
    <employee>  
        <firstName>Anna</firstName> <lastName>Smith</lastName>  
    </employee>  
    <employee>  
        <firstName>Peter</firstName> <lastName>Jones</lastName>  
    </employee>  
</employees>

**Web service code reference**

|  |  |  |  |
| --- | --- | --- | --- |
| **Number** | **Name** | **SOAP fault** | **Description** |
| 201 | Created | Server | The request has been fulfilled, resulting in the creation of a new resource. |
| 200 | OK | Server | The request has succeeded. The information returned with the response is dependent on the method used in the request |
| 301 | Moved permanently | Server | The requested page has been permanently moved. The server automatically redirects the request to the new location. |
| 304 | Not Modified | Server | The server has decided, based on information in the request, that the requested data has not been modified since the last request and so it does not need to be sent again. |
| 307 | Temporary Redirect | Server | The requested page has been moved, but this change may not be permanent. The server automatically redirects the request to the new location. |
| 400 | Bad Request | Client.BadRequest | The HTTP request is incomplete or malformed. |
| 401 | Authorization Required | Client.Authorization | Authorization is required to use the service, but a valid user name and password were not supplied. |
| 403 | Forbidden | Client.Forbidden | You do not have permission to access the database. |
| 404 | Not Found | Client.NotFound | The named database is not running on the server, or the named web service does not exist. |
| 408 | Request Timeout | Server.RequestTimeout | The maximum connection idle time was exceeded while receiving the request. |
| 411 | HTTP Length Required | Client.LengthRequired | The server requires that the client include a Content-Length specification in the request. This typically occurs when uploading data to the server. |
| 413 | Entity Too Large | Server | The request exceeds the maximum permitted size. |
| 414 | URI Too Large | Server | The length of the URI exceeds the maximum allowed length. |
| 500 | Internal Server Error | Server | An internal error occurred. The request could not be processed. |
| 501 | Not Implemented | Server | The HTTP request method is not GET, HEAD, or POST. |
| 502 | Bad Gateway | Server | The document requested resides on a third-party server and the server received an error from the third-party server. |
| 503 | Service Unavailable | Server | The number of connections exceeds the allowed maximum. |