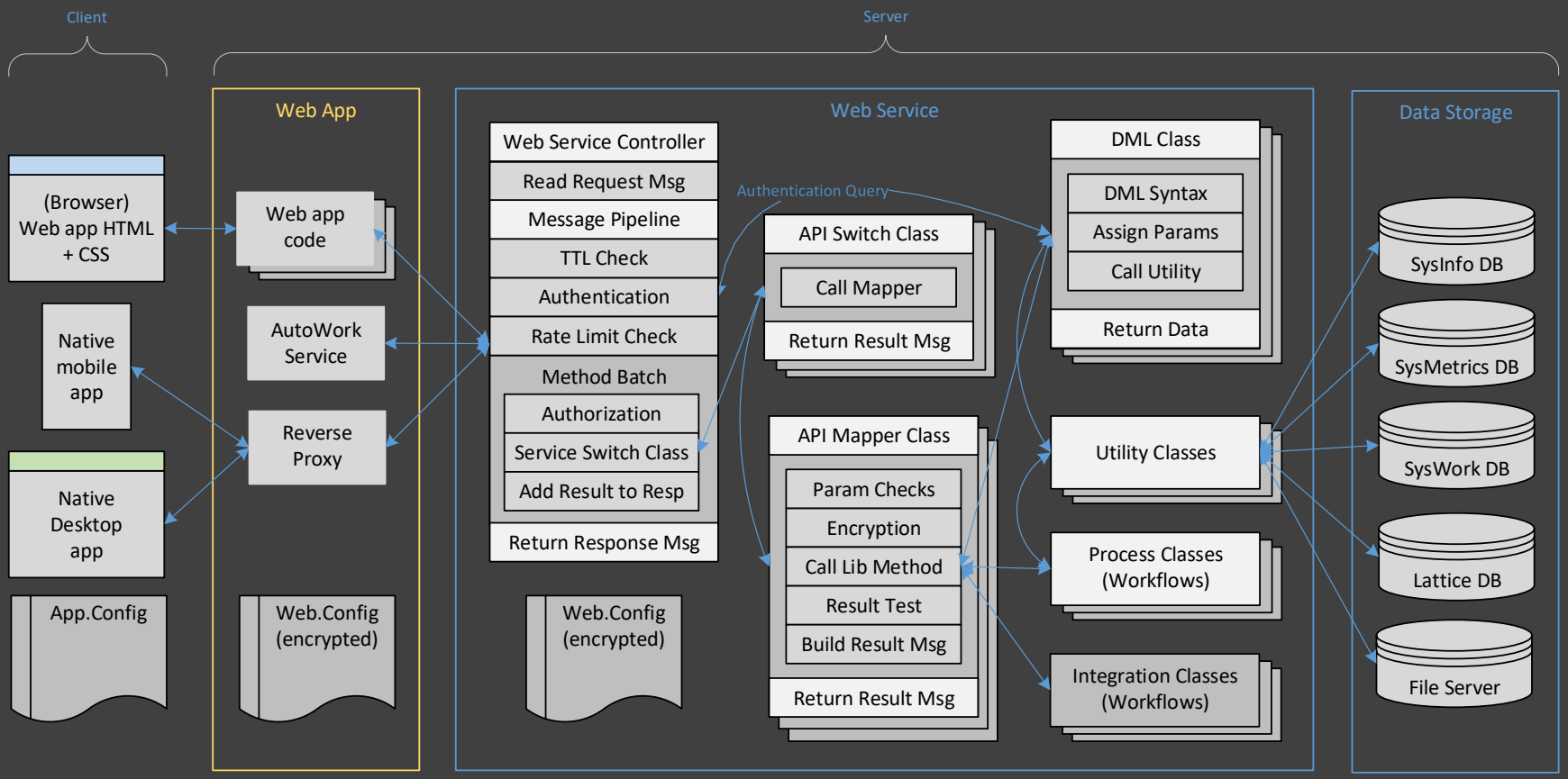
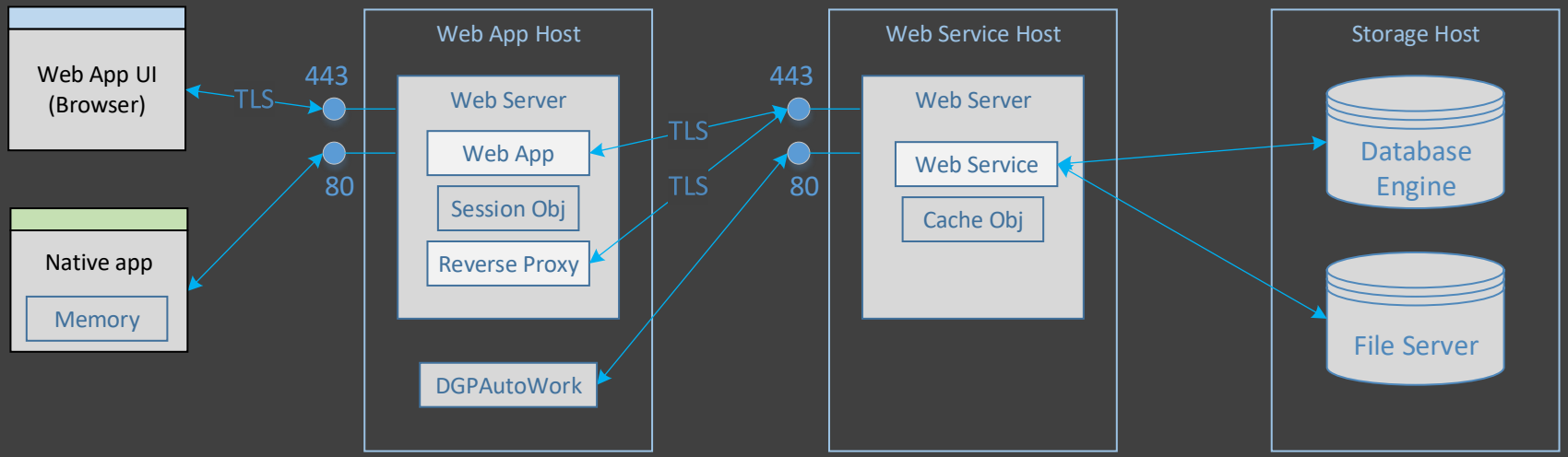


DGP Lattice uses web applications for the documentation web portal and reverse proxy pages. If a location has scaled up to the point that it has a DMZ (for network security), then web applications are deployed in the DMZ, along with the reverse proxy and the AutoWork service app to run replication processes between locations. For production locations, the firewall rules will hopefully forbid any software running on the internal network from making any direct calls to resources outside of the internal network. For this reason, replication processes must run in the DMZ where calls to external locations are permitted.

Web Apps

The overall architecture of DGP consolidates all of the logic in a system into the web service API's, and then focuses on constantly making improvements to the security, performance, availability, reliability, scalability and testing of those web services. In that context, client applications contain little if any logic themselves, acting more as a thin presentation layer that handles user events and calls API methods for all app functionality. This centralized hub-and-spoke architecture provides many benefits, but is only feasible if the web services are very fast, with end-to-end round trips between the client and the server completed in well under 1 second. While there is nothing to force applications to be designed and built to follow that convention, the overall systems will generally work much better when they do.



In some cases, the DGP reverse proxy pages will provide the necessary functionality for a system, but that will be determined by the use of TLS, where TLS is terminated, and whether or not load balanced server farms are being used in the DMZ and Internal network. If the reverse proxy pages do not meet the requirements, some other reverse proxy hardware or software will need to be used.

TLS

TLS is used to encrypt the API request and response messages during transport, but it is not needed for all environments. The use of TLS or HTTP to communicate between the tiers is handled by the web server and is transparent to the web apps and web services, with no differences in their functionality. Establishing TLS connections are an expensive operation, so reusing those connections once they have been established is important for good performance. Internally, either TLS or HTTP connections can be used, depending on what level of security is needed for a specific system.

Caching

Both native apps and web apps cache information about the user and the web service they are connected to. The native app caches that data in its own local memory. Static data is stored in the app.config file of the native app, which is read into memory when the app is initialized.

DGP web apps use Server Side Rendering (SSR), and cache the same data as native apps in the session object (memory) of the web app server. Web services can optionally also cache information in the cache object (memory) of the web service server. This caching functionality built into Windows and the .NET Framework eliminates the need for a caching subsystem such as Redis to be added to the system architecture. This type of distributed caching solution works equally well for a single server, small scale systems, and larger systems, and is generally going to be much more scalable and fault-tolerant than a Redis or other type of cluster – however, centralized, global caching tools can be used in DGP systems as needed.

Field Name	Field Values	Description
LocState	ONLINE, OFFLINE	A state value used to enable/disable the web app
SvcURL		The URL of the web service used by the web app and reverse proxy