Architecture Models

CRM.COM is based on a modern scalable event based micro-service architecture with cloud or on-premise hosting using Kubernetes. The architecture follows a Domain Driven Design pattern and uses CQRS techniques and event management to ensure maximum performance, scalability, flexibility, extensibility and functional features are all well met.
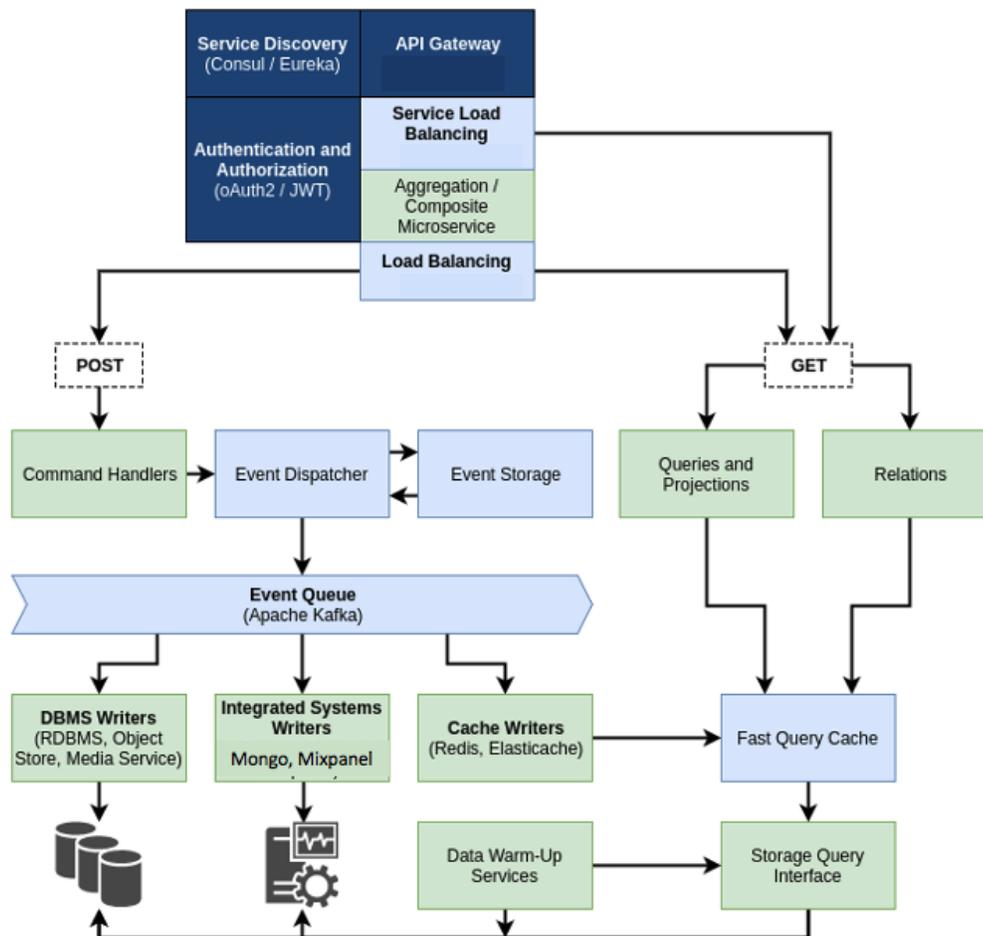
The server components utilize multiple domain based micro-services, coupled with event handlers that allow a number of intensive operations to be managed in an asynchronous approach improving performance of the user experience and managing load effectively on the servers.

CRM.COM API / Domain Design

CRM.COM follows best practice with a domain defined set of micro-services. Each domain has its own service and where necessary has aggregation to other services by using the REST APIs of those services.

CRM.COM is built on an event based architecture allowing business processes to cross domain but also to allow maximum extensibility via our webhook or automation configuration.
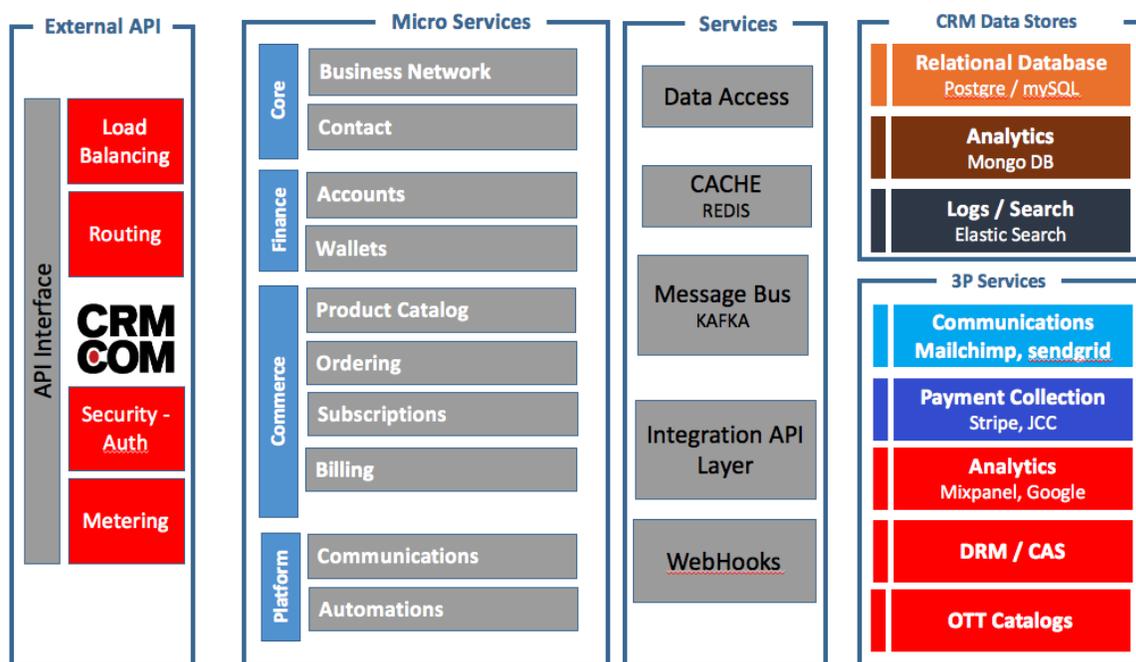
Key Technologies

1) **API Gateway:** The API gateway is responsible for the ingress of requests from the external load balancers and the routing of API calls the correct microservice or aggregation service. All services also separate write operations from read operations and the gateway is responsible for this routing.

2) **Kubernetes**: All microservices are managed and hosted in cloud environments using Kubernetes.

3) **REDIS**: For high speed cache of data that is used frequently the platform utilizes REDIS memory cache, with persistence storage to minimize warm up.

4) **KAFKA**: To manage the event based architecture the platform posts and consumes events on KAFKA. This architecture is also key to third party integrations like provisioning adapters or analytics.

CRM.COM API / Integration

CRM.COM provides maximum integration and extensibility through the event based architecture. As a business process is executing, such as placing an order for a number of services, the events in that flow can be exposed via webhooks, CRM.COM's pre-built adapters or third party adapters meeting our specification.

Multiple events can be connected to automations or webhooks. For example, a new customer registration can be connected to a communication platform such as Mailchimp and then be sent to Mixpanel.

A number of pre-built adapters using third party APIs are also available to allow plug and play integration to other leading platforms:

1) **Communication** – Mixpanel, Sendgrid
2) **Analytics** – Mixpanel, Google
3) **DRM** – Verimatrix, Fairplay, Widevine
4) **CAS** - Nagra
5) **OTT** - Netflix

CRM.COM API / Web Services

CRM.COM has a fully documented API split into two sets of APIs, to allow an integrator to build either administration or web / mobile app functionality.

The APIs are REST based, provide JSON response and follow the CRM.COM process models.

API	Description	URL
Self Service API	Used by Mobile APP's and web portals and design to support either CRM consumer Authentication or external oAUTH authentication services	https://speca.io/CRM/self-service
Back Office API	Used by the CRM Back Office web platform and designed to support either CRM Admin User Authentication or Server to Server Secret Key	https://speca.io/CRM/backoffice-admin

CRM.COM Security

The CRM.COM security has two parts:

1) Backoffice users - BackOffice
2) Mobile apps and web portals – Self Service

Both use industry standard JWT tokens for each of the relevant APIs where the JWT token and a refresh token are provided by the relevant authentication methods.

CRM.COM also supports oAUTH authentication from various external servers such as Facebook. The oAUTH token can be provided via an authentication API. The

CRM.COM servers will validate the token with the oAUTH servers and if successful they will swap to a CRM.COM token for subsequent API access.

In addition, for secure server to server communication, the CRM.COM backoffice APIs support a secret key mechanism that can be created and updated using the backoffice tool. These secret keys allow API use without the need for a server to "login" and ensure secure access that does not require holding "JWT token state".

Backoffice functionality also extends the token to provide "roles" that a user ID configured for these roles provide access permissions to the APIs. So, for example a finance API can be secured based on a role and will restrict access if that user does not have the roles in the token.

For more information or a demo contact us at: info@crm.com