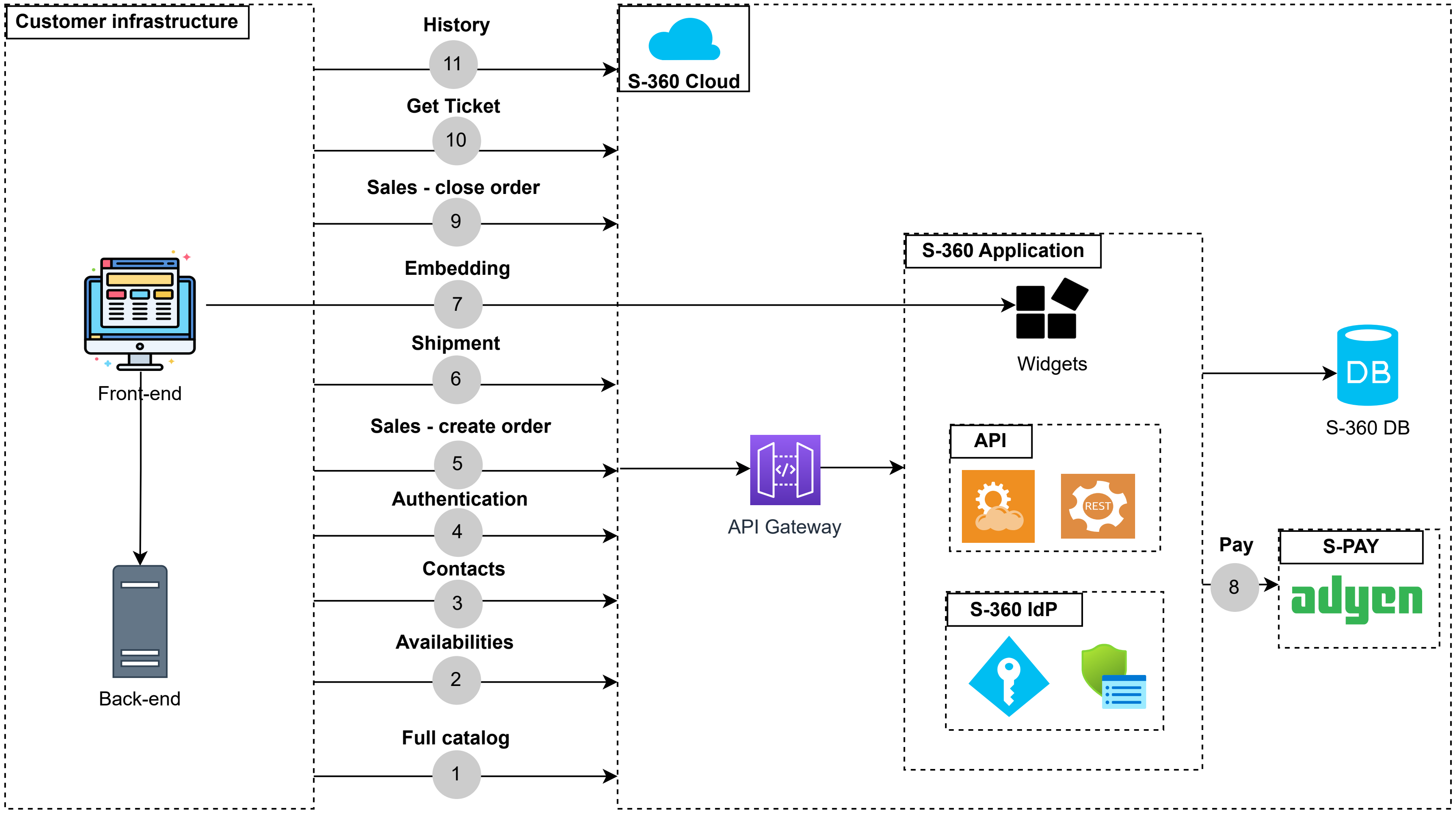
Reference Architecture diagram for website full integration

The Reference Architecture diagram illustrates how the SecuTix platform facilitates full integration, enabling seamless integration.

In this scenario, you have the freedom to create a fully customized ticketshop experience using only our APIs. From catalog display to ticket purchase and payment, every step can be tailored to your specific requirements. This architecture helps you to build a ticketing solution that aligns precisely with your brand and business needs.



1 **Full Catalog** : S-360 Catalog API, allows you to retrieve the full catalog. This includes detailed product descriptions, such as names, dates, tariffs, prices, and sales periods. Given that using the catalog API can be expensive and is subject to rate limits, it is crucial to implement an effective caching strategy and then just call **getLastSetupUpdate** (Catalog). For further information, please refer to the catalog backend API page.

2 **Availabilities** : S-360 offers an API for retrieving the catalog & availability updates. These availabilities are categorized using color codes. The availability service offers data access through four key methods:

- **getProductAvailability**,
- **getEventAvailability**,
- **getPerformanceAvailability,**
- **getSeatsAvailability**.

Typically, these are best used hourly, with a maximum frequency of 1 call per minute, particularly leveraging **getUpdatedAvailabilities** for monitoring product changes. To maintain data freshness while respecting API limits, implementing caller-side caching is recommended. You can find more information on the availabilities API pages.

3 **Contacts:** Administer your contacts with ease using S-360's API. The **ContactInformationService** allows you to efficiently create, update, and delete contacts, among other functionalities. Associates can easily link Contacts with addresses and set criteria. Access our Contact API pages for user-friendly documentation.

4 **Authentication** : Numerous APIs are offered by the S-360 application to streamline authentication. While S-360 can act as an Identity Provider (IdP), activation of this function requires specific configurations. For insights into Single Sign-On (SSO) processes, consult the "S-360 as IdP" page.
Additionally, integration with third-party IdPs as a Service Provider (SP) is smoothly facilitated, guided by a well-structured architecture reference.

5 **Sales - create order** : Initiating a sales workflow is simplified with S-360's APIs, aiding in the creation and closure of orders.Starting by checking product availability with the Availability API (done in Step 2). then proceed to retrieve the payment method ID through the **GetPosConfig.** Following these steps, order creation is executed using the **ExternalOrderFacade** web service.
Explore the sales pages for more details and practical examples.

6 **Shipment** : S-360 provide different shipment methods tailored to each contact's address, accessible through a variety of shipment modes provided by S-360. Use the **ShipmentPublicService** web service for a range of solutions to effectively manage shipments linked to an order. Additional information is available on the Shipment pages.

7 **Embedding**: Appropriately embedding S-360's payment widget with the IDs from step 5 is crucial. Adequate configuration is imperative for integrating the widget into the ticket shop, ensuring continuous PCI compliance. Access to documentation for widget integration is provided upon request.

8 **Pay :** Previously integrated, the Widget initiates communication with S-PAY, a vital component of S-360's offerings. Designed to handle operation with the Adyen solution, it ensures that once payment is completed and the transaction validated.

9 **Sales - close order** : To finalize the order, the payment need to be successfully completed in step 8, then the utilization of the **ExternalOrderService** API is required, it will finalize the order and the customer is redirected to their website.
Further details and illustrative examples can be accessed on the sales pages.

10 **Get ticket:** Detailed management of ticket access and shipment requires the **ProductionPublicService** web service, especially important for activities like ticket printing.

11 **History:** Secure access to a contact's order history is facilitated through the **OrderHistoryService** web service, crucial for accessing details pertaining to all orders completed for each contact.

All the API definitions and specifications, can be found in the Secutix platform website.

Reviewed for technical accuracy January, 2024

Secutix Architecture Reference