



# Integration Guide

Integration with ai:Go has been designed to be as easy as possible but due to the varied nature of deployments, options have been provided to accommodate instances from single camera to full smart city.

By the end of this document you should be able to answer the following questions, What:

- Deployment type (SaaS / Embedded)
- Camera Connectivity (Public / Firewall exceptions / Peering)
- Camera Protocol (RTSP / Other)
- Camera IPs to connect
- Data output (Batched / Streaming / Integration)
- Batched Output (Email / Cloud / SFTP)
- Realtime Format (msgpack / protobuf / raw json)
- 3rd Party Integrations to include

## **Deployment**

There are two main types of deployment

- SaaS (Software as a Service)
- Embedded

*SaaS deployment is always preferred unless prohibited by security constraints.*

## **SaaS Deployment**

## ai:Go System Architecture



### **Data Input**

Allowing greater flexibility and scaling a SaaS offering also allows extremely quick setup. For camera feeds exposed publicly before commencement with ai:Go, congratulations! Your system is ready to begin collection.

Provide a list of IPs to collect from, ai:Go's collection servers will then attempt to connect and begin processing data automatically.

ai:Go connects via RTSP. There is flexibility to collect and process other formats or authentication mechanisms.

If there are security firewalls in place, we can provide an IP to whitelist. If there are more complex peering requirements get in contact to work through this integration.

### **Data Storage / Redundancy**

ai:Go keeps no image data longer than needed. Once it is processed the data can be cleared. This is desirable as the amount of data generated from a single camera can exceed 100 GB per day.

Track data is retained however contains no identifiable information or information that is able to trace back to an identification once the image data is cleared. This can be fine tuned for privacy or security reasons.

### **Data Output**

To receive data from ai:Go a chosen method (or methods) of data collection need to be specified.

Data output comes in three types, batched output, streaming output and via an integration. There is no limit to the number of data output options e.g. Integration into a third party system can be accompanied with a batched daily export.

#### **Batched**

Batched output consists of files that are periodically exported to a chosen location. For example a CSV file of 5 minute traffic count data on a daily basis.

Batched output can be stored directly on a customer cloud account, emailed, SFTP transfer or made available from a dropbox account.

The frequency, batch interval and transfer options are left for the customer to decide.

### *Realtime*

ai:Go can provide real time streaming data with sub second latency. This can be provided directly as a data stream, socket or via a webhook.

Data formats for real time streaming can be msgpack, protobuf and raw or binary encoded json (with or without compression)

The data exported can be all tracking data or vehicle exit data. All tracking data can be extremely intensive for even a single intersection and subject to updates as the tracker continues to resolve paths in real time. Logic is required to process these messages to ensure there is no double counting.

*Note: Email for the message schema for the desired data format.*

### *Integration*

The aim is to provide automatic integrations to all mainstream tools. The list of integrations is always expanding as we look to provide data in a multitude of formats. Each integration has a one time cost associated with setup and maintenance costs.

Email for costing and a setup guide for a specific integration.

*Note: Data Richness will be affected by what output is chosen. For the richest data use a real time stream. Integration data input will be restricted by support from the developers of the software.*

### **Embedded**

If security constraints prohibit allowing remote access to the data streams, embedding within a customer datacenter is an option.

Due to scalability and load sharing, SaaS deployment is preferred. It may be uneconomic to deploy an embedded instance collection less than 10 cameras full time.

There is also no ability to apply multi-tenancy discounts based on hardware usage (*the system will be billed at capacity for its life cycle*).

A minimum install will require 11U of server space.

Additional capacity and redundancy can be added as needed. This is dependent on requirements and may or may not require additional rack space.

SSH connectivity to the system is still highly desirable for remote management and updates. If this can't be provided additional cost will be incurred for periodic updates and maintenance.

Provisioning will still follow the black boxed topology as the SaaS topology however an internal connection will be maintained instead of an open firewall port.

Data output may be limited by security constraints if there is no option for external connectivity. Batching to an internal file store is an option but will need to be developed to suit.

Third party integrations that collect data may also be unreachable without a firewall exception. Once again this is dependent on the environment and how strict the security requirements are.

**Reach out**

If you have further questions, integration support or custom requirements reach out via email [support@ai-go.nz](mailto:support@ai-go.nz)