Trauma App

Product & System Description

Document Control

Version	Date	Author	Role	Changes Noted
~~1.0~~	~~16.12.21~~	~~James Tulloch~~	~~Project Manager, Daysix~~	
~~1.1~~	~~26.05.23~~	~~Tom Montgomery~~	~~Head of Operations, Trauma App~~	
~~2.0~~	~~02.05.24~~	~~Tom Montgomery~~	~~Head of Operations, Trauma App~~	
~~2.1~~	~~08.05.24~~	~~Tom Montgomery Ben Beaumont Mac Beagan~~	~~Head of Operations, Trauma App. Head of Software, Daysix. Lead Developer, Trauma App.~~	
~~2.2~~	~~21.01.25~~	~~Tom Montgomery~~	~~Head of Operations, Trauma App~~	~~Update to Section 3.11~~
2.3	19.02.25	Tom Montgomery	Head of Operations, Trauma App	Adjustment for Option B Daysix-hosting

Contents

1. Introduction

1.1 Context

Trauma, the fourth leading cause of death in western countries and the leading cause for people under 40, has a significant economic, social, and environmental impact.

Trauma care, a complex set of interventions and investigations driven by a structured clinical exam, is currently delivered by a team with a leader and a scribe for documenting the case.

The Trauma App aims to streamline this process by replacing paper forms and retrospective note-taking, tracking the full trauma case, and capturing and monitoring interventions in real-time. It seeks to improve adherence to the Advanced Trauma Life Support (ATLS) protocol, an international standardised approach to trauma care, addressing the acknowledged variations in trauma care.

1.2 Definitions

In relation to this document:

"Company" or **"Supplier"** or **"Daysix"** means Daysix Medical Ltd, the owner and licensor of the software.

"Hospital" or **"User"** or "**Customer"** means the team of clinicians using the software in the hospital's emergency department and associated health board who are the purchasers as described in the contract.

"App" means the software application as supplied by the company as described by the Product Description in section 2 of this document.

"NMTR" means the National Major Trauma Registry which is NHS England's audit governing body for national trauma data auditing and review, which recently replaced The Trauma Audit & Research Network (TARN).

"ATLS" means Advanced Trauma Life Support, which is a method used by medical professionals to immediately assess and treat injured patients.

'FHIR" means 'Fast Healthcare Interoperability Resources' which is the global industry standard for passing healthcare data between systems.

"ED" means the Emergency Department of the hospital where major trauma is treated.

"EPR" means the hospital's Electronic Patient Record which is used within healthcare practices to easily create, store, send and manage patient records, sometimes known as the Electronic Medical Record (EMR).

2. Product Description

2.1 Overview

The app is an iPad application designed for Major Trauma Care clinicians in a Hospital Emergency Department (ED) to digitally document patient care in major trauma cases. The app allows for data collection through an iPad App user interface. Data is stored directly on the device for offline use and regularly synced to the server to ensure continuity in case of device failure. Upon case closure, a senior clinical user verifies and signs off the documented data, which is then transferred to the server and deleted from the local device.

The app enables documentation of a patient's major trauma care in hospital from Standby (pre-call alert) and team preparation prior to patient arrival through to the patient being moved into the broader ED patient pathway (e.g. sent to Theatre or ICU for ongoing monitoring). The app covers the full spectrum of assessments, investigations, interventions and procedures provided in the trauma resuscitation bay and has bespoke forms which allows those documenting (Scribes) to maintain a high level of real-time quality data in a fast-passed clinical setting.

When the patient moves into the ongoing care pathway, upon senior clinical sign-off, a high fidelity PDF case report is generated and/or structured data (e.g FHIR) is pushed to the hospital digital records. Case history and user administration is accessible through a web based dashboard.

2.2 Value Proposition

10 Key Features:

1) Robust, time-stamped, real-time data documentation at the point of care
2) Capture data rapidly with iPad mobility, proven more usable than paper.
3) Offering decision support, with checklists and visual prompts
4) Uses familiar care frameworks and following ATLS protocol.
5) Enables standardised automated reporting of Major Trauma Care.
6) Hand-held, mobile, with multi-device same case input functionality.
7) Captures more comprehensive data than existing processes.
8) Integrates with the hospital's existing systems, pushing and pulling data.
9) Shows data in Power BI analytics format for real-time clinical review.
10) Displays data in familiar formats (e.g. NEWS/PEWS observation charts).

10 Key Benefits:

1) Enables closer analysis of the clinical care processes.
2) Reduces variation and standardises care, improving patient outcomes.
3) Enhances accuracy of reporting to Trauma audit bodies.
4) Improves access to any available uplift tariff for high performance.
5) Reduces administration requirements saving time and money.
6) Reduces length of stay by improving pathways of care.
7) Real-time metrics against KPIs, surfacing Power BI tools, enhancing analysis.
8) Displays prompts and checklists, helping clinicians deliver safe and consistent care.
9) Provides learning opportunities for clinicians with access to performance data.
10) Identify and spread examples of excellent practice.

For Clinicians:

- **Helps them perform at their best -** Active decision support at the point of care increases their ability to deliver safe and consistent care (especially helpful when nearing the end of long shifts where tiredness is a proven issue). The Trauma App actively evaluates all data points and reviews against the ATLS protocol to lightly prompt clinicians with excerpts/references when relevant.
- **Time saving / Reduce admin -** Automated reporting, reducing paperwork and validation to ensure completeness. (No case admin after the clinical episode, no illegible handwriting, reduction in follow-ups from colleagues and no loss of key data.)
- **Live performance data** - Surfacing of appropriate key performance indicators provide instant validation of the care delivered. Additionally, metrics provide a growth target and a means to analyse personal performance.

For Patients:

- **Improved care -** Optimised care delivery in the Emergency Department, increased adherence too checklists and reduction in variation.
- **Reduced morbidity and mortality**
- **Enhanced rehabilitation pathways**

For Hospitals:

- **Increased revenue -** Increased accuracy of documented procedures, prescriptions and tests. More acute billing to insurance companies and increased uplift from national audit bonus schemes (e.g National Major Trauma Registry (NMTR) within NHS England).
- **Cost savings -** Reduced requirement for administrative staff to collate and prepare trauma documentation.
- **Reduced patient stays** - Improved patient outcomes leads to reduction in ongoing bed usage for major trauma patients.
- **Improved documentation** - Vastly improved case documentation, improving communication of patient travelling through hospital departments while affording better protection from litigation.
- **Simplified national reporting** - Improved adherence and reporting to national standards with much lower administrative footprint.
- **Real time performance metrics** - Review overall hospital performance against national KPIs in real time. (Existing paper systems see hospitals looking at data that is 6 months behind making it difficult to evaluate the overall impact of optimisations)
- **Identify weaknesses early** - Live data feeds enabling offering rich insights to help identify strengths and weaknesses and facilitate rapid evolution of practice.

For Boards & Trusts:

- As above for hospitals.
- **Performance evaluation across the entire network** - Compare hospital performance across the network to identify where support is needed early.

2.3 Evidence

Following conception and development in partnership with the 4 Major Trauma Centres in Scotland spanning a 4 year period, the App began use in live trauma cases in 2021 in NHS Scotland, NHS Glasgow and Greater

Clyde (GG&C) at the Queen Elizabeth University Hospital (QEUH). In 2023 Alder Hey Children's hospital was the first site in NHS England to begin using the App live in practice.

Existing Research Studies

> *The Trauma App supports checklist adherence, which has already been proven to support ATLS workflow - https://pubmed.ncbi.nlm.nih.gov/31392359/*

Published Research Studies:

> Usability & Data Completeness studies have been carried out on the App at the QEUH and NHS GG&C.
>
> Results show that digital real-time recording of clinical events using a tool such as the TraumaApp is comparable to completion of paper proforma. The System Usability Score for the Trauma App was above the internationally validated standard of acceptable usability.
>
> Links to these research papers can be found at thetraumaapp.com/resources

Planned Research Studies:

> QEUH and NHS GG&C are in the planning stages for evidencing the administration time saving impact of reporting to the Scottish Trauma Audit Group (STAG).

Future Research Studies:

Clinical teams using the App are being encouraged to support further study to evidence:

- Speed to key KPI's (e.g. Time to CT, Time to TXA, Time to Antibiotic)
- Reduction in Error
- Audit body uplift tariff impact as it relates to NMTR Uplift Tariff.

3. System Description

The app has been designed to offer a high quality documentation system that works seamlessly within existing clinical practice.

The app's system has been designed to work in all hospital environments, including those *without an internet connection*. Where an internet connection is available the case can be worked on by multiple devices with data syncing between devices in fractions of a second. Data is tagged with user ids providing full traceability.

3.1 Data points (Archetypes)

Every touch of a button generates a discrete datapoint containing all the data required to trace the action and build a structured record. These are called archetypes (Analogous to a FHIR message).

An archetypes is represented by a series of classes that inherit from one another to create a final archetype:

Base Class	Name	A identifier to represent the archetypes
	CreatedTime	Timestamp that captures to milliseconds when the datapoint is created
	EventTime	A timestamp that captures when the scribe believes the event took place
	Context	The application context in which the datapoint was created. E.g: app/standby or app/standby/physiology
	UserID	The ID of the user (If logged in) who created the event
Type Class	Type	The type of the archetype. E.g. it could be a number, single select list, multi select list, time, etc.
Archetype Class	Properties	Properties that are specific to that particular archetype

The benefit of generating a discrete datapoint for every screen click is that it's entirely transparent. The datapoint contains everything needed to identify its origin and speaks to its own authenticity. For example, if the app timestamp was an hour after the event timestamp it can confidently be said that this datapoint was added retrospectively. The reverse is equally useful allowing it to be said that the data was captured in real time where they match.

3.2 Archetype Queue

The app adopts a non destructive data model. A case is represented by a queue of events which is only ever added to. Consider a queue of events containing an archetype representing age.

Case Start

Consider a case where the user receives a phone call stating that the incoming patient is roughly 40 years old. Initially they miss type it as 44, notice their mistake and correct it to 40. On arrival, the patient has been identified and we now know they are in fact 39.

In a destructive data model, this value would simply be overwritten each time and previous values lost. With a queue based non destructive model all the data is retained. The final report will show the patient age as 39, but the queue has the capability of telling the full story.

The non destructive model provides a complete audit trail of changes to the clinical record at a data point level. It also has a useful side effect of making multi user concurrent case access simple and entirely safe. Any number of iPads can connect to a case and add data points.

3.3 Offline support

The non destructive queue model also facilitates offline or patchy connectivity by caching locally the queue and syncing whenever a connection is made available. An entire case can run offline and maintain a local queue.

3.4 Structured readable data

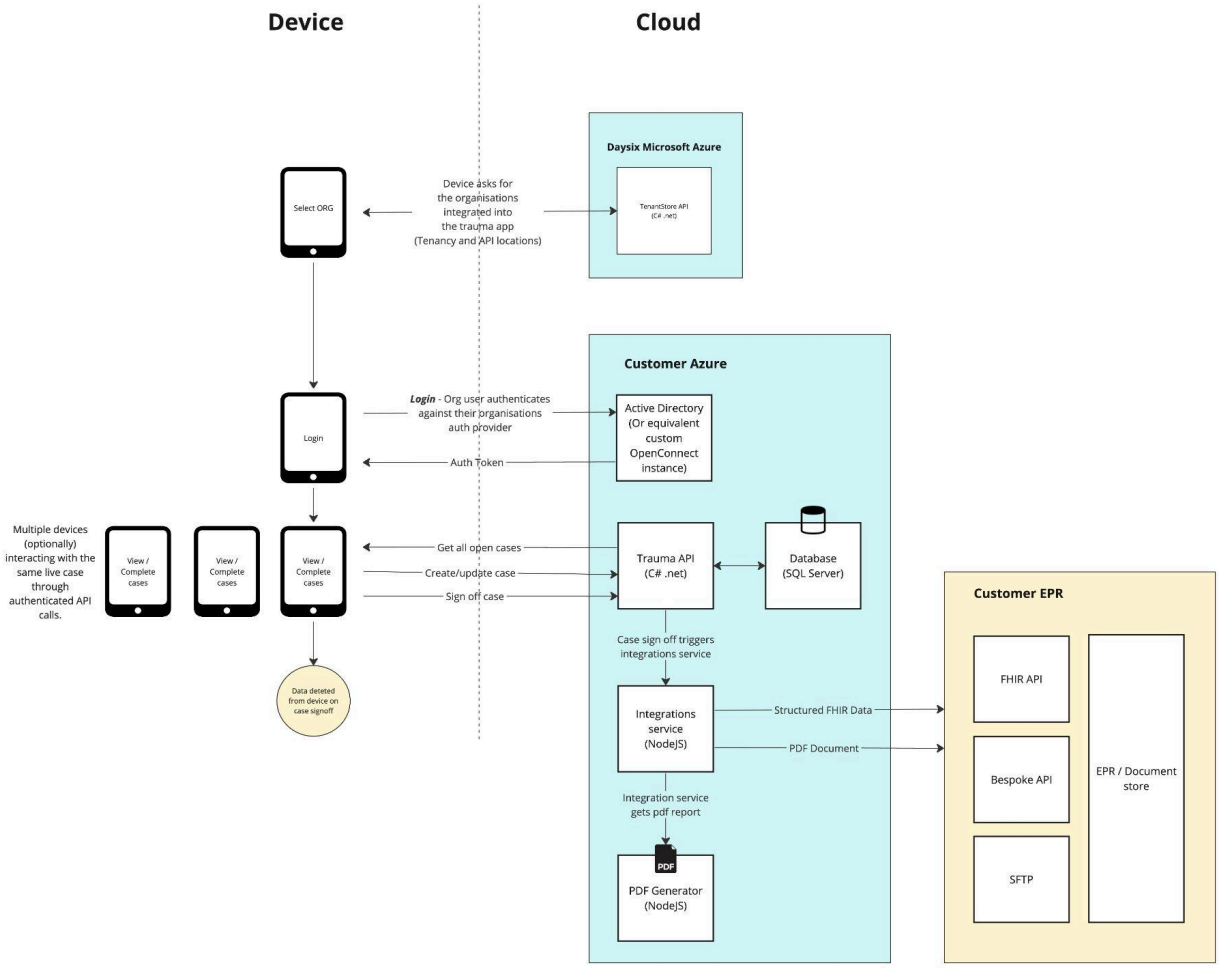
The one downside of a queue of discrete events is that they can't be easily interpreted. The addition of a *context* to each event allows the queue to be transformed into a structured record (JSON/XML/FHIR etc) that is easily read and interpreted.

3.5 System Interoperability

This concept of an archetype closely resembles FHIR, OpenEHR and other medical formats. As a result, a case queue can be easily transformed into a FHIR data stream for good interoperability

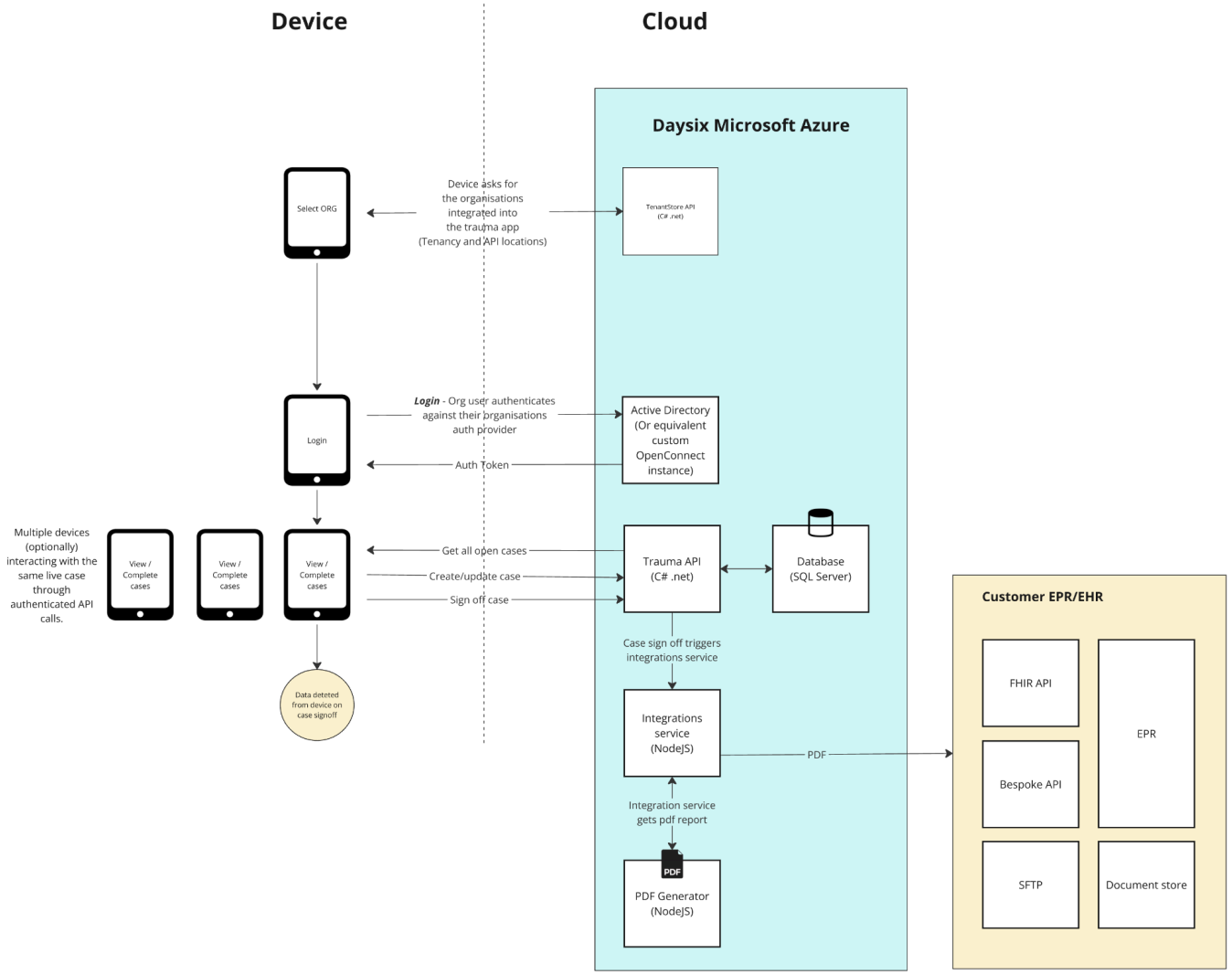
3.6 System architecture and Data Flow (Customer hosted - Option A)

The diagram below illustrates the flow of data around the core building blocks of the Trauma App and its cloud services. The key design principle of our solution when choosing to self-host (Option A) is that the data never leaves your deployed ecosystem and is always entirely in the customers control.



3.7 System architecture and Data Flow (Daysix hosted - Option B)

The diagram below illustrates the flow of data around the core building blocks of the Trauma App and its cloud services if Option B (Hosted by Daysix) is chosen which is the preferred route for running a pilot of the app before full production deployment with Option A Installation (3.6). In some cases with relevant governance approval, Option B can be used for Live Deployment post-pilot, within a full procurement implementation.



3.8 Key components

Application (ReactNative)	The core application is written in ReactNative. It was chosen to allow us to not only support multiple mobile platforms, but ultimately allows us to port for the web in future. It provides the ideal flexibility allowing us to target the systems that best suit our users. iPads were chosen as the first target platform because of their perfect form factors for fast accurate data entry. For safety, the case only remains on the device while active. Upon completion, the entire case is removed completely from the device once a connection can be established to the server. As a result, the iPad is merely a data collection tool and there is minimal risk if the device is lost.
Dashboard (Angular)	A companion dashboard provides clinicians with a secure portal to retrospectively review cases.
API (C#)	Both the iPad and dashboard connect to an Azure hosted API through which all communication with the data is conducted. Users must be authenticated before they can read or write data. This is a code first C# .NET solution based on the latest version .NET 8 (as of April 2024) and latest best practices.
Multi Tenant Authentication	The trauma ecosystem can be accessed via a customer's authentication mechanism of choice. The API and database architecture has been designed to facilitate a multi-tenant approach where data is separated into discrete databases for each customer. The default authentication mechanism is Azure Active Directory (AAD)
Data store (MS SQL Server)	Our datastore of choice is MS SQL Server. We have chosen to use a completely managed service and implemented using .NET Core code first approach. The benefit of this approach can not be overstated. Data integrity is incredibly important to us, so adopting tooling that allows us to build, evolve and rollback is exceptionally important. Scalability is also important and this approach allows us to spin up additional discrete servers in minutes.
PDF generator / Report service	Availability of information is critical in a health setting so our design sees us extract the report generation and distribution into discrete services. This provides redundancy by design ensuring that any failure to create and push the PDF into the clinical system fails well and tries again. The report service continues to attempt to generate and send the PDF in ever increasing intervals, sending failure reports at each point. This provides immediate visibility of issues, while protecting the integrity of the data while engineers fix the issue. Once patched, all missing reports will be generated

	and pushed.
Push to clinical system	Our API and report service are set up to 'push' to a clinical record using FHIR messaging. With appropriate security credentials and data sharing agreements, we can push the data in a report or structured form to a customer's agreed destination.
Pull from our system	Where a customer prefers to pull data they can do so via our API once authenticated. They can also register for a notification when a case is signed off allowing them to pull on demand rather than poll.

3.9 Delivery

Our technical engineering team delivers changes using an automated multi-staged environment configuration. All changes are committed to a code repository via pull request. These changes are automatically built, tested and deployed into our first environment called 'development' for internal evaluation. If accepted, those changes are then promoted systematically through a series of environments and tested at each stage before reaching production.

Development	The first environment used to test internally by the development team.
Integration	The first environment used for testing by our core clinical advisory group made up of a small group of clinicians who lead the project. All data in this environment is simulated data.
User acceptance testing (UAT)	The second environment available to all clinicians who use the application. They can evaluate and test the new features before they enter production. All data in this environment is simulated data.
Production	The live environment delivered to customers

3.10 Technologies

ReactNative	The core application is built in ReactNative, a javascript framework for building cross platform native applications. It has become a mainstream technology which is used by companies like Microsoft, Meta, AirBnB and Tesla.
Angular	The companion dashboard that allows clinicians to sign in to find and view retrospective cases. It is written in Angular, another well used Javascript platform and developed and maintained by Google. Angular was chosen because it was

	designed specifically to build this kind of dashboard that consumes data of a secure RESTful web service.
MS SQL Server	All application data is stored in MS SQL Server databases. This technology was chosen because it is the technology of choice for NHS Scotland.
C# REST API	The API through which the application accesses data is written in C# .NET core using a code first approach. The technology provides strong patterns that support the effective delivery of a secure, highly performant and scalable API.
AAD Authentication	The application, API and dashboard all require Azure Active Directory authentication.
Azure Hosting	The API and all associated databases are hosted on MS Azure. This provides robust security features out of the box while also allowing the application to scale depending on the need. Additionally, each customer can host their own dataset providing clear separation of concerns and data siloed appropriately. In Scotland (UK), the API and database is hosted on NHS Scotland's own Azure infrastructure, allowing the entire product to run safely within their SWAN network.
Azure DevOps	We leverage DevOps to create a multi environment development pipelines, allowing us to support a multi staged rollout of application changes. Rolling back or cherry picking changes can also be performed incredibly quickly.

3.11 Hardware requirements

a. **Servers**
The exact specification of cloud infrastructure will vary per customer. However, the requirements for the backend services are reasonably light. The primary load on the servers is from the signal-r implementation that streams data between devices.
Worked Example for comparison: NHS Scotland hosts the Trauma App cloud services centrally for all of Scotland. At peak this will see around 5000 major trauma cases per year. The services run well on a 2 core, 8GB RAM, 250GB Storage virtualized instance (e.g. MS Premium v3 P1V3 Azure plan*). SQL Server usage is light allowing this to run on an Azure Standard S1: 20 DTUs 250GB Storage*. As a guide, for both production and development MS Azure instances this will cost in the region of £500 p/month.[1]

[1] MS plans offered on 08.01.2025 and subject to change. See - https://azure.microsoft.com/en-us/pricing/

b. **On Premise Hosting**
The trauma cloud services can be deployed to a managed host like Azure, or virtualised using Docker to run on premise. Both options can make use of our automated build pipelines allowing any/all customers to receive the latest updates automatically. When hosting on premise, we recommend running two instances of the cloud services to allow for 100% uptime as deployments take place. For example using Rancher.

c. **iOS Devices**
Apple iPad only with an A12 processor or higher. Essentially any standard iPad after the 8th Generation (2020) and supporting the latest version of iPadOS.

d. **Device Management**
Customers are encouraged to at the very least enable automatic updates on their devices. This ensures the device takes critical application updates. Most customers prefer to use a management service like Airwatch, to push the latest app updates onto the device.

e. **Charge Management**
It's critical that the devices are kept charged to ensure availability. Docking stations in the resus bay has become the standard approach by most customers.

4. Integrations

4.1 Types

There are 4 types of integration that can be offered to users:

a. **Authentication**
Utilises your existing identity management system allowing staff to login using credentials they are familiar with.

b. **Patient Identity/Personal Demographics Service (PDS)**
Verify patient details through a lookup service (e.g. CHI Lookup in Scotland, or NHS Number via the Spine network in England). The Trauma App supports identifier lookup from demographics or demographics from patient identifier.

c. **EPR Document repository**
Push PDF case reports into the hospital's EPR.

d. **EPR Direct**
Push structured data into the hospital's EPR. Pull physiology, interventions, notes, results etc. from the hospital's EPR.

4.2 Authentication

The Trauma App can be configured to authenticate with any OAuth2/OpenConnect compliant provider.

4.3 Patient Identity/Personal Demographics Service (PDS)

The Trauma App can be configured to perform lookups against your patient identity services. This is flexible, with the data request and response payloads translated by the Trauma API. As a result request and response payloads can be in JSON, XML, FHIR and take advantage of SOAP or REST.

4.4 EPR Documents

The base integration between app and EPR sees the final report (PDF) pushed in the EPR document store and surfaced in the EPR. This takes place immediately on case sign-off ensuring that onward care teams immediately have access to a comprehensive record of events in the emergency department. Documents can be sent to another API via a REST call (JSON/XML/FHIR) or uploaded over SFTP.

4.5 EPR Direct (Structured Data)

Trauma App -> EPR

Structured data can be pushed directly to your EPR. A complete record of all events (or a subset) can be pushed at case signoff to the EPR. Where required, data can also be pushed in real-time. For example, imaging requests, physiology etc. We can collaborate with customers to build integrations that suit their EPR systems and data formats.

* The table below is using the app terminology rather than their FHIR counterparts

Observations	Respiration Rate Oxygen Saturation Heart Rate Blood Pressure Temperature GCS
Event	Time of injury Time of arrival Mode of transport Source of transport Mechanism of injury Broad areas of injury
Medical History	Allergies Known Medications Significant Comorbidities Last ate Last drank
Examinations	Primary Survey Secondary Survey (Injuries) ASIA Pupil Exam
Imaging requests	CT MRI XRAY
Point of care tests	Blood Gas Blood Glucose COVID ECG Pregnancy Thromboelastogram (Rotem) Ultrasound Urinalysis Blood labs (Biochemistry, Haematology etc)

Transfusions	Blood products Fluids
Prescriptions	Medications * Ventilator Settings
Interventions	Cardiac Arrest (2 minute drill) Active Rewarming Airway Adjunct Arterial Line Cannula Central Line Chest Compression Chest Drain Cleaning (Wounds) Compression C-Spine Protections Defibrillation Dessings Facial Packing Femoral Split Fracture Reduction Glue IO Join Reduction Laparotomy Lateral Canthotomy Nasal Packing Needle Decompression Nerve Block Pelvic Splint Plaster Resuscitative C-Section RSI Staple Surgical Airway Suture Temperature Maintenance Thoracostomy Thoracotomy Tourniquet Transfer Device Urinary Catheter
Case Details	Staff / Team Notes Movements Disposition Outstanding Items Next of kin notes Child protection summary

* The app is not a prescribing solution. It is documentation of the timings and flow of care. Current customers use the pharmacy solution as their formal record of truth.

EPR -> Trauma App

Your EPR can also push data into a trauma case. Data will be accepted if a live trauma case can be found that has a patient identity that matches the incoming payloads. Supported messages include:

Patient Identity	ID Surname Given Name Date of birth Gender
Medical History	Allergies Known Medications Significant Comorbidities Pregnancy
Observations	Respiration Rate Oxygen Saturation Heart Rate Blood Pressure Temperature GCS
Point of care test results	Blood Gas Blood labs (Biochemistry, Haematology etc)
Prescriptions	Ventilator Settings
Other	Staff / Team Notes