

## First4Aid Application

### Context

As governments around the globe struggle to support their populations in destabilised regions, non-governmental organizations (NGOs) have stepped in to provide life-saving aid (UNHCR, 2022). One key problem faced by both aid providers and aid requestors is the ability to securely communicate (Harper and Dobrykowski, 2022; Loy, 2022). Whilst encrypted applications such as Signal and WhatsApp have emerged as point-to-point options, they do not allow the create, read, update, delete (CRUD) operations required to effectively share data within an NGO to provide the requested aid (Loy, 2022). Because of this, NGOs have been using platforms and practices that fail to safeguard vitally sensitive information such as the names, locations, and health information of vulnerable populations (Harper and Dobrykowski, 2022; Loy, 2022).

### Scope

The scope of this application, First4Aid, focuses on the core ability to facilitate secure communications between aid providers (NGOs) and aid requestors (people in need). This application will allow a requestor to send a message to a provider, a provider to respond to a requestor, logging to enable activity audits, and secure data storage that adheres to standards within the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), and ISO 27000. This use case is diagrammed in **Appendix A**. The scope of this application does *not* cover the business practices or tradecraft outside of this application: for example, it will not help a provider determine the veracity of information coming from a requestor or provide insight as to how aid should be provided to a requestor.

### System Requirements and Assumptions

First4Aid is designed for large-scale humanitarian crises, such as those observed in Afghanistan and Ukraine. Each of these events has displaced massive populations, with millions of people at acute risk of starvation, exploitation, or violence (UNHCR, 2022). This scale derived our technical assumptions, to the right, and the following system requirements.

#### Technical Assumptions

- First4Aid will support 100,000 users
- Average user will require 300MB for travel documents and multimedia
- 30TB total storage requirement
- Storage does not account for archives or historical data, only active users
- Initial First4Aid proof of concept will operate at a fraction of this scale

Compute resources will emphasize stability and availability over performance, maximizing the number of processors, cores, and threads (Pillai, 2017). Humanitarian crises are most resource-intensive during their onset – so implementing an elastic compute solution that can surge to handle an influx of requests during the first several weeks, then scale down as the situation stabilises, is critical for both cost and performance considerations (UNHCR, 2022). For this reason, First4Aid will leverage Amazon Web Services (AWS) Elastic Compute Cloud (EC2) to provide performance when needed and scale down when the load decreases (Amazon Web Services, 2019). First4Aid will need to handle concurrent data streams from thousands of aid requestors, and queue transactions from multiple users on the same documents to preserve data integrity and availability.

Due to the geographically disbursed nature of NGOs and their aid recipients, and the desire to leverage cloud-based resources, all access to this system will be remote. First4Aid will have three types of users, listed in **Table 1**.

Table 1: First4Aid's Users and Permissions

User Role	Create	Read	Update	Delete
<b>Aid Requestor</b>	Initial aid request message; response messages to Aid Providers	Messages sent to and from the user	Messages sent by the user	Messages sent by the user
<b>Aid Provider</b>	Response messages to Aid Requestors; messages to other Aid Providers	Messages sent to and from the user	Messages sent by the user, assign requests to a case	Messages sent by the user
<b>Aid Provider Admin</b>	Messages to any user, cases, database records	All messages, cases, database records	All messages, cases, database records	All messages, cases, database records
<i>All actions will be logged for auditing purposes</i>				

## Design

First4Aid prioritises security and functionality through a simple, purpose-built design. Key features include:

- Event-Driven Architecture to manage the communication between the various software containers that compose the app (shown in **Appendix B**)
- Zero Trust Architecture (ZTA) to authentication and authorisation of users (Rose et al., 2020)
- Low-bandwidth user interface built using Flask and Django frameworks, accessible via computer or mobile device (shown in **Appendix C**)
- Web application written in Python using object-oriented programming design principles (shown in **Appendix D**)
- Message queueing and event monitoring using RabbitMQ and Datadog, respectively (RabbitMQ, n.d.; Datadog, n.d.)
- asyncio Python module to enable concurrency across multiple threads (Pillai, 2017)
- SQL database for storage, enabling data encryption in transit and at rest

## Anticipated Security Challenges

The vulnerabilities from the Open Web Application Security Project (OWASP) Top 10 List in **Table 2** present the greatest risk to First4Aid based on its design and technologies used.








Table 2: Anticipated Security Challenges, Mitigations, and Paradigms (Mend.io, n.d.; OWASP, 2021; )

Vulnerable Aspect	OWASP Security Risk	Mitigation/Paradigm
<b>User roles/permissions</b>	A01:2021 Broken Access Control	Apply principle of least privilege; deny by default; enforce record ownership
<b>Encryption, safe data transfer, data storage</b>	A02:2021 Cryptographic Failures	Implement HTTPS for all internal and external transmissions; discard unnecessary sensitive data; store log-in data using functions like Argon2
<b>Login, data entry, cross-site scripting, SQL inject</b>	A03:2021 Injection	Validate input; proper escaping of input data; use SQL functions such as LIMIT to minimise amount of data potentially released

Vulnerable Aspect	OWASP Security Risk	Mitigation/Paradigm
<b>Data transmission across all components</b>	<i>A04:2021</i> Insecure Design	Perform threat modelling; integrate security language in requirements and design documents; limit resource consumption by user/service; leverage defence in depth paradigm
<b>Server operating system, database, web application</b>	<i>A06:2021</i> Vulnerable and Outdated Components	Log component versions, perform updates as early as practicable, check vulnerability databases for known issues
<b>Login, web application front end (Flask/HTML)</b>	<i>A07:2021</i> Identification and Authentication Failures	Impose strong password requirements; limit or delay consecutive login failure attempts; implement multi-factor authentication

### Tools and Libraries

First4Aid development will leverage a variety of tools and libraries, including those listed below.

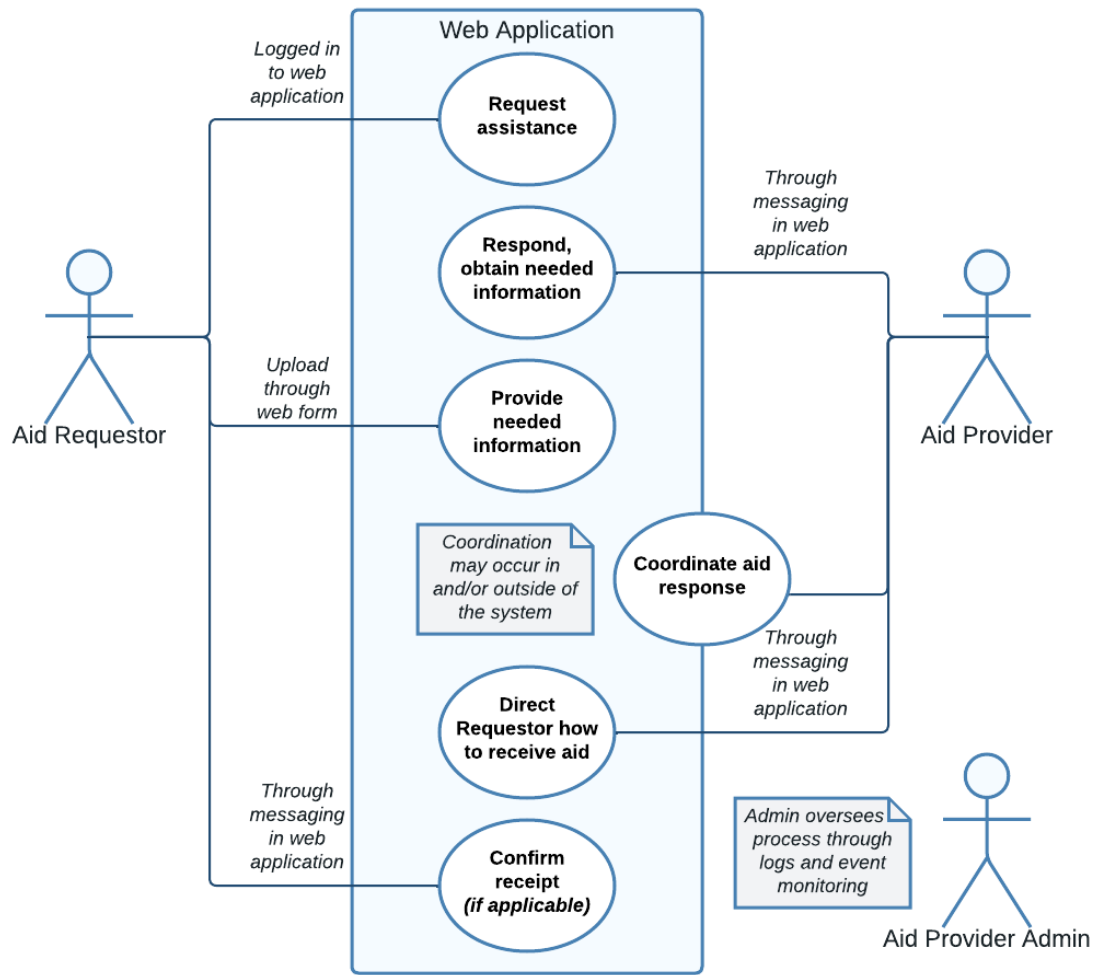
	<b>Datadog:</b> Software-as-a-Service that provides event monitoring for web applications and infrastructure (Datadog, 2021)
	<b>Docker:</b> Operating system-level virtualisation to enable event-driven architecture and containerisation of system components (Docker, n.d.)
	<b>GitHub:</b> Code repository for development, collaboration, version control, and documentation (GitHub, n.d.)
	<b>PyCharm:</b> Integrated Development Environment (IDE) for application development (JetBrains, n.d.)
	<b>Jupyter Notebook:</b> Notebook-style IDE allowing for rapid, iterative, manual code tests (Jupyter, n.d.)
	<b>Python unittest library:</b> Functions that automate unit testing procedures in Python (Python Software Foundation, 2022)
	<b>RabbitMQ:</b> Open-source Message Queuing system supporting authentication and authorisation (RabbitMQ, n.d.)

## References

- Amazon Web Services (2019). *Amazon EC2*. [online] Amazon Web Services. Available at: <https://aws.amazon.com/ec2/>. [Accessed 20 Jul. 2022].
- Datadog (2021). *AWS Monitoring | Datadog*. [online] AWS Monitoring. Available at: <https://www.datadoghq.com/solutions/aws/> [Accessed 23 Jul. 2022].
- Docker (n.d.). *Enterprise Application Container Platform*. [online] Docker. Available at: <https://www.docker.com/>. [Accessed 23 Jul. 2022].
- GitHub (n.d.). *GitHub*. [online] GitHub. Available at: <https://github.com/>. [Accessed 23 Jul. 2022].
- Harper, N. and Dobrygowski, D. (2022). *Why the humanitarian sector must make cybersecurity a priority*. [online] World Economic Forum. Available at: <https://www.weforum.org/agenda/2022/01/why-humanitarian-sector-cybersecurity-a-priority/> [Accessed 23 Jul. 2022].
- JetBrains (n.d.). *PyCharm*. [online] JetBrains. Available at: <https://www.jetbrains.com/pycharm/>. [Accessed 23 Jul. 2022].
- Jupyter (n.d.). *Project Jupyter*. [online] Jupyter.org. Available at: <https://jupyter.org/>. [Accessed 23 Jul. 2022].
- Loy, I. (2022). *'It's like the wild west': Data security in frontline aid*. [online] The New Humanitarian. Available at: <https://www.thenewhumanitarian.org/interview/2022/02/28/data-security-in-frontline-aid> [Accessed 23 Jul. 2022].
- Mend.io (n.d.). *Most Secure Programming Languages*. [online] Available at: <https://www.mend.io/most-secure-programming-languages/> [Accessed 20 Jul 2022].
- OWASP (2021). *OWASP Top Ten*. [online] Owasp.org. Available at: <https://owasp.org/www-project-top-ten/>. [Accessed 20 Jul. 2022].
- Pillai, A. (2017). *Software architecture with Python: design and architect highly scalable, robust, clean, and high performance applications in Python*. Birmingham, England: Packt Publishing.
- Python Software Foundation. (2022). *unittest — Unit testing framework — Python 3.8.2 documentation*. [online] Available at: <https://docs.python.org/3/library/unittest.html>. [Accessed 23 Jul. 2022].
- RabbitMQ. (n.d.). *Messaging that just works — RabbitMQ*. [online] Available at: <https://www.rabbitmq.com/>. [Accessed 23 Jul. 2022].
- Rose, S., Borchert, O., Mitchell, S. and Connelly, S. (2020). National Institute of Standards and Technology (NIST). *NIST Special Publication 800-207: Zero Trust Architecture*. [online] doi:10.6028/nist.sp.800-207.
- UNHCR (2022). *Global Report 2021*. [online] *UNHCR Global Report 2021: The stories behind the numbers*. United Nations High Commissioner for Refugees (UNHCR). Available at: <https://reporting.unhcr.org/globalreport2021/> [Accessed 23 Jul. 2022].

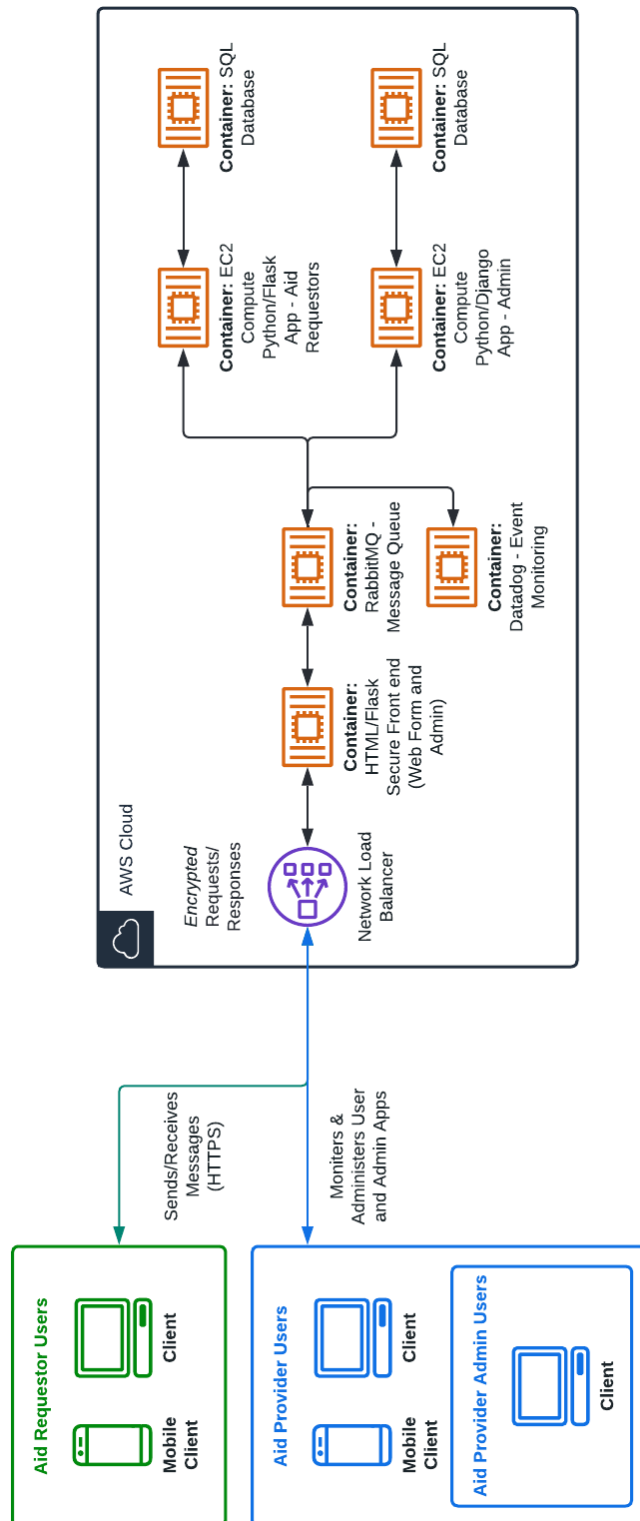
## Appendix A

### Use Case Diagram



## Appendix B

### System Architecture



## Appendix C

## Aid Request Web Form Design

## Request Form

Full Name\*

Phone Number\*

Type of Request\*

Location\*

No. of individuals requesting aid

Date Required\*

More information\*

Submit

## Appendix D

### Class Diagram

