BITBOX: A web-based application to transfer medical imaging data

Rubaida Easmin, Giovanna Nordio, Alessio Giacomel, Mattia Veronese
Centre for Neuroimaging Sciences, Institute of Psychiatry, Psychology and Neuroscience, King's College London, London, UK;

email: team-bitbox@kcl.ac.uk

Table of content

Highlights

The exchange of medical data supports multisite collaborations and boosts medical research. Usually, medical imaging data has immense volume, dynamic and diverse complexity, and they require special secure systems to be transferred between parties that protect individual privacy as well as data integrity. Despite many digital innovations, there are many technical and regulatory bottlenecks that make medical imaging data exchange challenging. Therefore, we have developed Bitbox which provides a reliable yet straightforward service to securely exchange (but not limited to) medical imaging data.

An Overview of Bitbox

Bitbox is a web-based application system that allows to transfer data with minimal technical difficulties in a secure way. With Bitbox, both imaging and non-imaging data can be transferred from any external and independent site into a centralized server (Figure 1). The application can handle any type of medical imaging data, including standard data format like DICOM [1], Analyze or Nifti [2] but not exclusively. The transferring system can be installed either as an on-premise solution, using in-house IT infrastructures, or as a cloud-based solution. For the latter applicative use of Bitbox, cloud computing technology Amazon Web Service (AWS)[3]has been used, since this guarantee on-demand resource availability, less maintenance and cost-effective services [3].

Figure 1: Overview of Bitbox. With Bitbox, data can be transferred from multi external and independent sites into a centralized server.

Workflow

The workflow of Bitbox is illustrated in Figure 2. Any user must be associated with an imaging centre or clinical site before using Bitbox, which must be preregistered within the Bitbox system. When the centre is registered and approved, it will be identified by a unique ID (1-3). Using this ID, the user can create his/her own Bitbox account (4-5). To activate this newly created account, the user needs to verify his/her email used for account registration (6). Once the account is active, the user can log in to the website and start uploading the files. In accordance with General Data Protection Regulation (GDPR) [4], any data must be anonymized before being uploaded in the system, thus it is the user's responsibility to verify that no identifiable information are contained in the data (8). When uploading the data, the user can specify the study ID and/or subject ID, select the project, and imaging modality (9-10). This information will help the data organization and further identification in the database.

When the data arrives at the server-side, the reviewer, who could be the admin user or any other user with granted access (researcher), will check the quality of the data (11-12). If the data are not properly anonymized or corrupted, they are rejected and removed from the system, and an email is sent automatically to inform the sender about the data rejection and the reasons for the rejection (13-14). On the contrary, if the transfer is successful, the data are downloaded and irreversibly removed from the Bitbox database. In fact, Bitbox behaves like a pre-archive system: once the data are sent and downloaded in the local network, they are removed from the database and no data record is kept in the system, while a transfer record is kept.

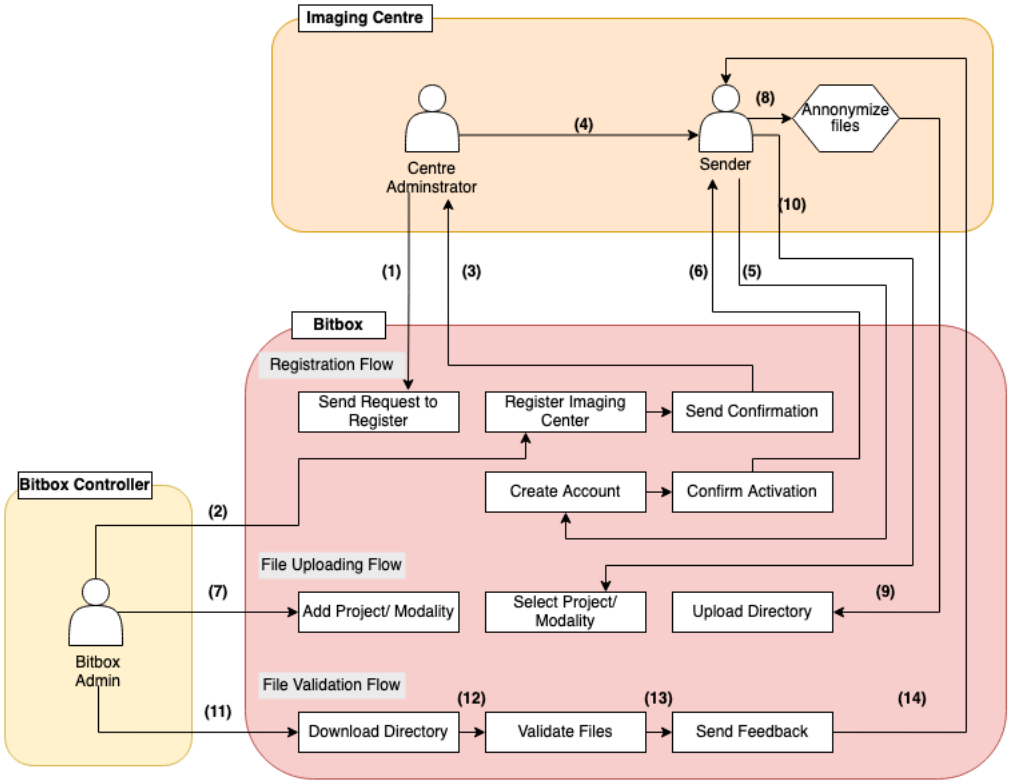

Figure 2: Workflow of Bitbox

The users can refer to the transfer records by going to the 'History' tab in the Bitbox application (Figure 3). Each record keeps some information about the transferred data and the respective sender, including the registration number of the imaging facility, the sender's ID, file size, date and status of the transaction. The status of a transaction indicates if the data have been accepted (*Success*), still under evaluation (*Pending*), or rejected (*Reject*) thus requiring to be sent again. Besides, the system admin can look for data sent from a particular imaging sites or on a specific date. It is also possible to export the whole record of the database as an excel sheet if required.

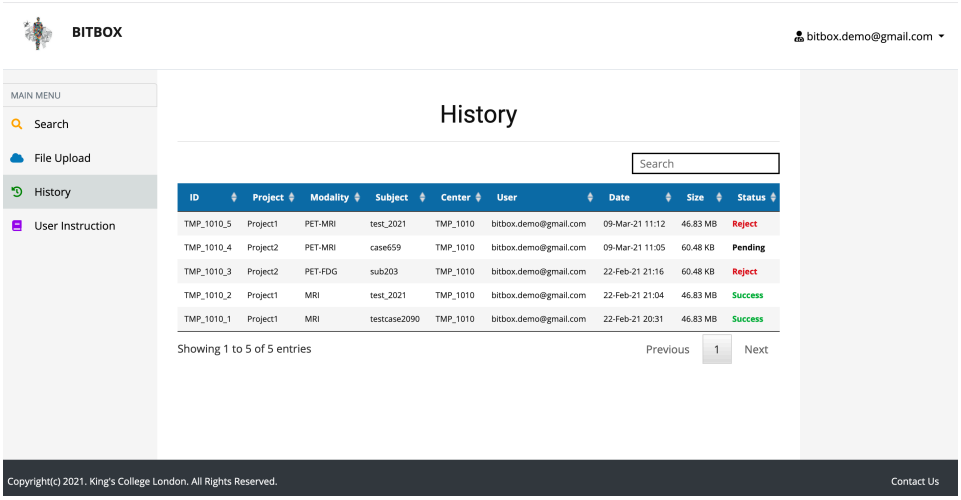

Figure 3: Bitbox web application: user interface of the 'History' tab with the record of all the user's data transaction.

System Architecture

The first version of Bitbox (https://www.bitbox-imaging.com/) has been created for internal use at the Centre of Neuroimaging Science, King's College London. However, the system is easy to configure, and can be customized for any particular research purpose. Bitbox architecture follows the widely used client-server architecture that includes a web-based user interface in app server, python-based middleware, a database server with relational database management system and a file server to store all incoming imaging files (Figure 4).

Bitbox consists of three main components, a registration system, a file uploader, and a management system. The application represents a fully operational data management system that organizes data based on the project, imaging modality, and the sender of the data.

The registration system is used to control users' access. Any user must be associated with an imaging center or clinical site, which has been pre-registered and approved by Bitbox admin. The user can access the system using his/her login credentials and the ID of the pre-registered clinical or imaging site he/she is part of. Only authorized users with an internet connection can access the application. This will help to manage data in one place as well as ensure data privacy and copyright issues.

Through the file uploader, data are transferred from the remote network of the imaging site into Bitbox. There are no restrictions on the data type that can be transferred, the only requirement is to upload the data stored in a directory. Generally, the directory can contain several files, but it is faster to upload a single large file instead of multiple small files due to the overhead of the transfer protocol. To speed up the uploading process, it is preferable to zip the files before the transfer. Uploaded files are stored in the file server. The system management module is managed by Bitbox administrative panel, which provides features to add a new study and imaging modality, and to register new imaging facilities.

While the user requests are received via the front-end part of the application, requests will be processed in the backend, which includes app server, database server, and file server. The app server hosts the whole system, which enables end-user interaction with the database server and file server through various protocol and application programming interfaces (APIs). The database server keeps the details of registration process as well as the regular

transactions information, whereas the file server stores documents and scans received from remote sites.

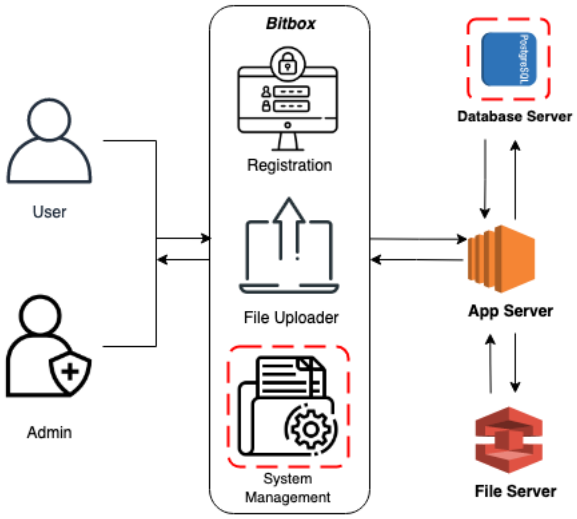

Figure 4: System architecture of Bitbox. The application consists of a front-end component (a registration system, file uploader and system management) to handle users' requests, and a back-end component (app –server, database server and file server) to process the users' requests.

Implementation and Deployment Details

Table 1 summarises the technology stack that has been used for the development of Bitbox. This system is built with Django, a free, full-stack, open-source, python-based web application development platform [5]. The architecture of Django follows Model-View-Template (MVT) pattern [3]. User-submitted requests from the website (Template) pass through the URL and routes each request to a 'View'. Then, the 'View' communicates to the database and transfers data to the 'Template' as a response. Additionally, some security features like Cross-site request forgery (CSRF), SQL injections, etc. are taken care of by Django. This strategy helps to minimize security vulnerabilities. A default Cascading Style Sheet (CSS) is included to enable a consistent look and feel for the overall site. It can be personalised to suit individual project aesthetics. Nginx server has been used as the web server for production. Bitbox also used PostgreSQL for its database backend.

Table 1: Web Development Stacks used for Bitbox

Technology Stack	**Name**
Framework	Django, Bootstrap
Programming Language	Python, HTML, CSS, JavaScript
Database	PostgreSQL
Server	Nginx

For the first implementation of Bitbox, AWS platform has been used since it ensures on-demand resource allocation, high-scalability and cost-saving services. Different AWS components have been used for the system deployment [3]. AWS Elastic Beanstalk facilitates the setup and configuration of the application, and it orchestrates other AWS components (EC2, RDS, Elastic Load Balancer, AWS S3). Besides Django default security, Web Application Firewall (WAF) rules are configured to increase security protection for the overall system. Bitbox is a closed-source application since the source code is not available publicly. On the basis of resource availability, it is also possible to deploy it in the on-premises network.

Data Access

Data stored in the Bitbox are restricted to access from public domain. In this first instance of the system, data can be downloaded only over King's College London (KCL) network, for storage and off-line processing. If the user is outside of the KCL domain, she/he must be connected via VPN before attempting to access the server. Moreover, the ability of individual users to download the data depends on the privileges assigned to them in Bitbox. The user account type is categorised into four different groups in the system: the imaging centre, the sender, the admin user and the researcher. The Bitbox system administrator is responsible for overall account management. It gives suitable permissions to different personnel and setup the new projects in the system. The imaging centre registration is a pre-requirement for any further user access into the system, but it does not have any direct access into the application. Registered users (*the senders*) are responsible for uploading data into Bitbox and they can only see the transaction that they have made in the application. The *admin user* can grant access to the admin panel to a particular user, the researcher, but with certain restrictions: the researcher can download the data from Bitbox but he/she cannot register new imaging sites and projects in the system.

Data Security & Privacy

In Bitbox, the application and database servers are kept separate so that any compromise on the application server cannot damage the information in the database. The database server stores and manages the data and provides data access for authorized users. This solution only allows SQL connection from the application server and no HTTP/HTTPS connection from the Internet. Further, a server certificate guarantees users the integrity and authenticity of the site and a Web Application Firewall (WAF) analyses web traffic and blocks any attempt to exploit vulnerabilities.

The Bitbox web application keeps the administrative and user panels separate for maintaining security protocols. Only Bitbox administrators can create and edit imaging facilities accounts, assign roles, activate or deactivate login privileges to the user accounts, and download data from the website. The administrative panel is well-protected by the configured firewall, which only allows clients to come from trusted/verifiable domains. Once the data is transferred to the permanent storage facilities, they will be kept in Bitbox until twelve months from the transfer. After that, they will be irreversibly removed from the server. However, data will be encrypted as long as they are in the application.

Although the data exchanged over the internet are encrypted, data must be pre-anonymized by the sender to ensure data confidentiality [6]. According to GDPR rules, it is an obligation of the sender to de-identify the personal data before transferring it to a third party[7] [8]. For additional security, data will be verified at the receiver end in Bitbox before approving or rejecting the transfer. It is, therefore, important that the controller will be familiar with the format and characteristics of exchanged data to perform the right assessment. If any data still contains identifiable information when received, they will be rejected immediately and removed from the Bitbox system.

Bitbox is compliant with GDPR regulation regarding the management of the users' personal information [4]. Following registration, the user agrees to share his/her personal information (name, email, id and workplace details). This is a primary requirement to determine the data source and the person behind the data transaction. Users can delete their accounts at any time and all his/her personal information will be removed from the Bitbox. The log of transactions made by the user will be kept in the system, but the information like sender's email, id and name will be anonymised. If the user wants to use the system in the future, he/she will require to complete again the registration with Bitbox.

System Validation

A complete test-suite based on Django's test-execution framework has been used to verify all functional specifications and system operation. The performance of the application is tested via an open-source performance testing framework named 'Locust' [9]. At present, the system is not expecting multiple client requests simultaneously, and the rising of incoming requests will increase the response time. Currently, the response time of file upload for a single user depends on the file size, number of files and the bandwidth of the network.

Bitbox has been deployed considering the fact that the initial number of uploads will not be more than 10 per week. Since the intended user request is relatively low, the IT infrastructure is configured accordingly. However, the application is provisioned to a load-balanced and scalable environment using AWS Elastic Load Balancing and Amazon EC2 Auto Scaling services. Therefore, the infrastructure can potentially handle a larger number of incoming requests. The database is also kept backup from time to time. In case of failure or problems with the database server, the system will move to another instance to continue operating without interruption.

The application currently allows uploading a directory with multiple files up to 2GB. To maintain the transaction integrity, files are stored in the AWS S3 bucket until notification of successful storage by the server. Moreover, individual file integrity is also checked during the transaction process via AWS file upload API. In case of internet disruption or file uploading failure, the database will not reflect any changes and the user will need to initiate the transaction process again. Compatibility testing [10] is also performed on the website to check if the application is compatible with a variety of browsers and devices.

Final remarks

The increasing rate of medical data has paved the way for medical research in diverse sectors - from drug discovery to disease diagnosis. Scientists are emphasising the democratization of the data more than ever. Legal issues like data security and protection, and information governance make such collaboration harder. Nevertheless, some research organizations have created their own platform to share resources and increase association among themselves. Majority of these solutions emphasize on the specific research area and have a particular work pattern. Exchange of data among communities is also diverse - either connected with a dedicated network (e.g.,VPN) or configure client endpoint. Researchers mostly prefer to concentrate on analysing data and solving the problems, rather than spending time collecting data from distinctive sites.

Bitbox has been presented to support data exchange in the field of medical imaging research. However, the application is a flexible system, and it can also be easily adapted for different research-oriented data transaction purpose rather than medical imaging. It is an easy-to-use web application where no additional configuration is required. Once registered with the system, it is ready to transfer data from the remote server to the institution's network, protecting the confidentiality and safety of the data.

Regulation & Disclosure

FDA considers a product to be a medical device, and subject to FDA regulation if it meets the definition of a device per Section 201(h) of the Federal Food, Drug, and Cosmetic Act (FD&C Act). As part of this regulation, "Software functions that are solely intended to transfer, store, convert formats, and display medical device data or results" are considered as Non-Device Medical Device Data Systems (MDDS) and are not subject to FDA regulatory requirements applicable to devices (that would include design controls) [11].

Bitbox is in alignment with Non-Device MDDS FDA policy, and with such definition it is not subject to FDA's regulatory oversight.

Limitation of Liability

Bitbox is not a certified product of any sort. To the extent permitted by applicable law, the creators of Bitbox will not be responsible for any damage, data loss, or whatever resulting from other people's utilisation.

References

[1] W. Dean Bidgood, S. C. Horii, F. W. Prior, D. E. van Syckle, and W. D. Bidgood, "Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging."

[2] M. Larobina and L. Murino, "Medical image file formats," *Journal of Digital Imaging*, vol. 27, no. 2. Springer New York LLC, pp. 200–206, 2014, doi: 10.1007/s10278-013-9657-9.

[3] "Amazon Web Services (AWS) - Cloud Computing Services." https://aws.amazon.com/ (accessed Feb. 03, 2021).

[4] European Union, "Regulation 2016/679 of the European parliament and the Council of the European Union," *Official Journal of the European Communities*, 2016.

[5] "The Web framework for perfectionists with deadlines | Django." https://www.djangoproject.com/ (accessed Feb. 03, 2021).

[6] L. Bradford, M. Aboy, and K. Liddell, "International transfers of health data between the EU and USA: a sector-specific approach for the USA to ensure an 'adequate' level of protection," *Journal of Law and the Biosciences*, Oct. 2020, doi: 10.1093/jlb/lsaa055.

[7] "Recital 26 EU General Data Protection Regulation (EU-GDPR). Privacy_Privazy according to plan."

[8] "Article 49 EU General Data Protection Regulation (EU-GDPR). Privacy_Privazy according to plan."

[9] "Locust - A modern load testing framework." https://locust.io/ (accessed Feb. 03, 2021).

[10] "Compatibility testing - Wikipedia."

[11] "Medical Device Data Systems, Medical Image Storage Devices, and Medical Image Communications Devices _ FDA."