triggre

Security, development and hosting

Contents

Introduction

The roots of our company lie in the custom software development of business critical systems which handle large volumes of transactions. The knowledge that our team has in that field, was all put into the processes we use for the development, security and hosting of the Triggre platform today.

We pay enormous amounts of attention to our development, security and hosting processes, to make sure that your application(s) and your data remain safe and secure at all times. This document describes those processes in detail.

Confidentiality

Due to the strategic nature of the content of this document, all of the information is strictly confidential and is only available upon request. It is strictly prohibited to provide this document and/or the information held in the document to anyone without the express consent of Triggre or a representative of Triggre.

Security

To provide a good overview of the security of the Triggre platform, we will first discuss the different roles and responsibilities within our team. This lays the foundation for a secure and auditable process.

From here on, when we talk about *production network* or *Triggre cloud* it means the servers containing the Triggre Designer, Triggre Builder and customer applications. When we talk about *company network* it means the computers we use for our daily work and the development and testing of Triggre.

Authorization

This authorization matrix contains an overview of who is responsible for and authorized to perform system level tasks. It excludes application level access, since that is up to customers themselves. It also excludes operations on the customers' application, since they are either responsible themselves (e.g. who is allowed to update the application) or is taken care of automatically by the Triggre platform (e.g. database migrations).

As a security measure, the systems of the Triggre platform are only accessible from within the company network.

Task	CTO	DRD	PM
Grant access to the company network	R / A	A	
Grant access to Triggre cloud management	R / A	A	A
Grant access to Triggre cloud servers	R	A	
Review updates and patch servers in Triggre cloud		R / A	
Roll out new versions of the Triggre platform		A	R / A
Troubleshoot customer applications when necessary		A	R / A

In the table the following keys and abbreviations are used:

- A - Authorized to perform task
- R - Responsible for task being performed
- CTO - Chief Technical Officer
- DRD - Director of R&D
- PM - Product manager

All other roles in the company have no access to the Triggre platform. Triggre Guides only have access to applications to help customers, but not to the underlying system. All development and testing environments are completely separated from the platform used for production purposes.

Note that the Director of R&D does not work on Triggre. Triggre development responsibility lies with the Architecture, Development and QA Leads, while the Director of R&D only facilitates the process.

Accounts

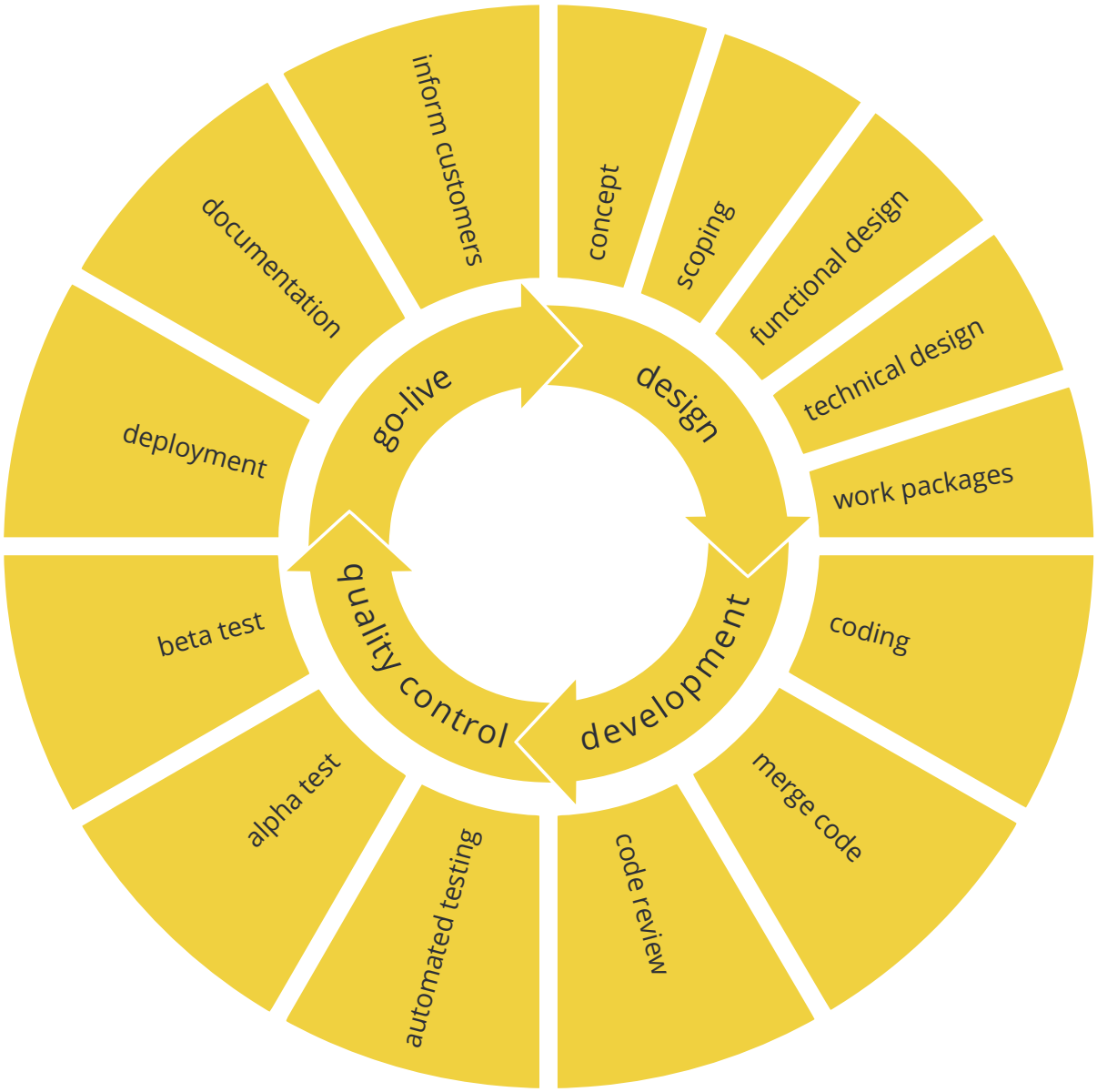
For database and server access we use personal accounts so we can log any changes to data. Only authorized personalized accounts which meet the authorization matrix have access.

Triggre applications have built-in audit logs to log all changes made to the application data, both changes made via the application by users and processes as well as changes made manually.

In practice, the Triggre Platform automatically takes care of all database changes (such as migrations from one version to another).

Development process

In this section we zoom in on the R&D process. The following diagram shows our development process with the phases in the smaller circle and sub phases in outer circle.



Design

In this phase, the design is made for new features that are going to be introduced in the Triggre platform.

The design phase consists of a number of sub phases:

- **Concept**
  In the first sub phase consists of the conceptual idea for a feature. The idea is conceptualized and written down in our documentation system.

- **Scoping**
  If a feature is very broad and involves many changes, we scope what parts of the feature will be developed. Other parts are placed on the backlog for later development.

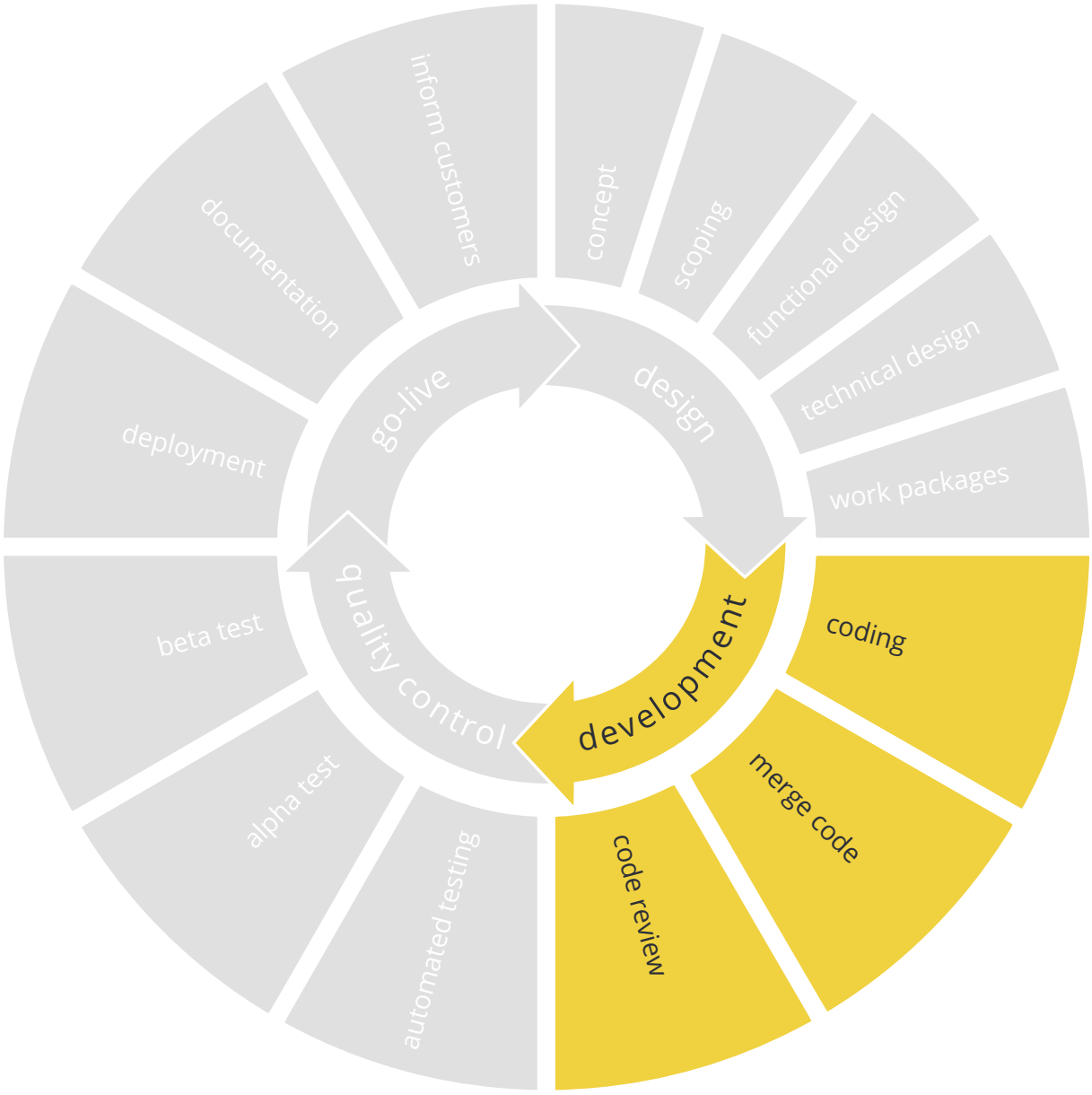
- **Functional design**
  When we have scoped a feature, we start with the functional design. The functional design describes what the user is able to do, what the screens look like and how the system should behave in specific situations.

- **Technical design**
  After the functional design is done, we continue with the technical design, which consists of any architectural changes to our system as well as guideline information on how a feature should be implemented.

- **Work packages**
  After the functional design is done, we continue with the technical design, which consists of any architectural changes to our system as well as guideline information on how a feature should be implemented.

Development

During the development phase, the actual source code writing process takes place.



The development phase is the most extensive of the R&D phases and consists of 3 sub phases:

- **Coding**
  When we code a feature, the new code is always developed in a separate 'stream', to prevent any unfinished features or fixes from entering any release prematurely. The coding of our system is further sub divided into steps, due to the complexity of the Triggre platform:

  1. **Designer front-end**
     The Designer consists of a front-end and back-end component. Most functionality we come up with are new components the user can use in his application. To do this, we must make the functionality available in the front-end of the Designer.

     Often this is in the form of, for example, a new action type or page type that the user can use, or a new option the user has available for an action or page.

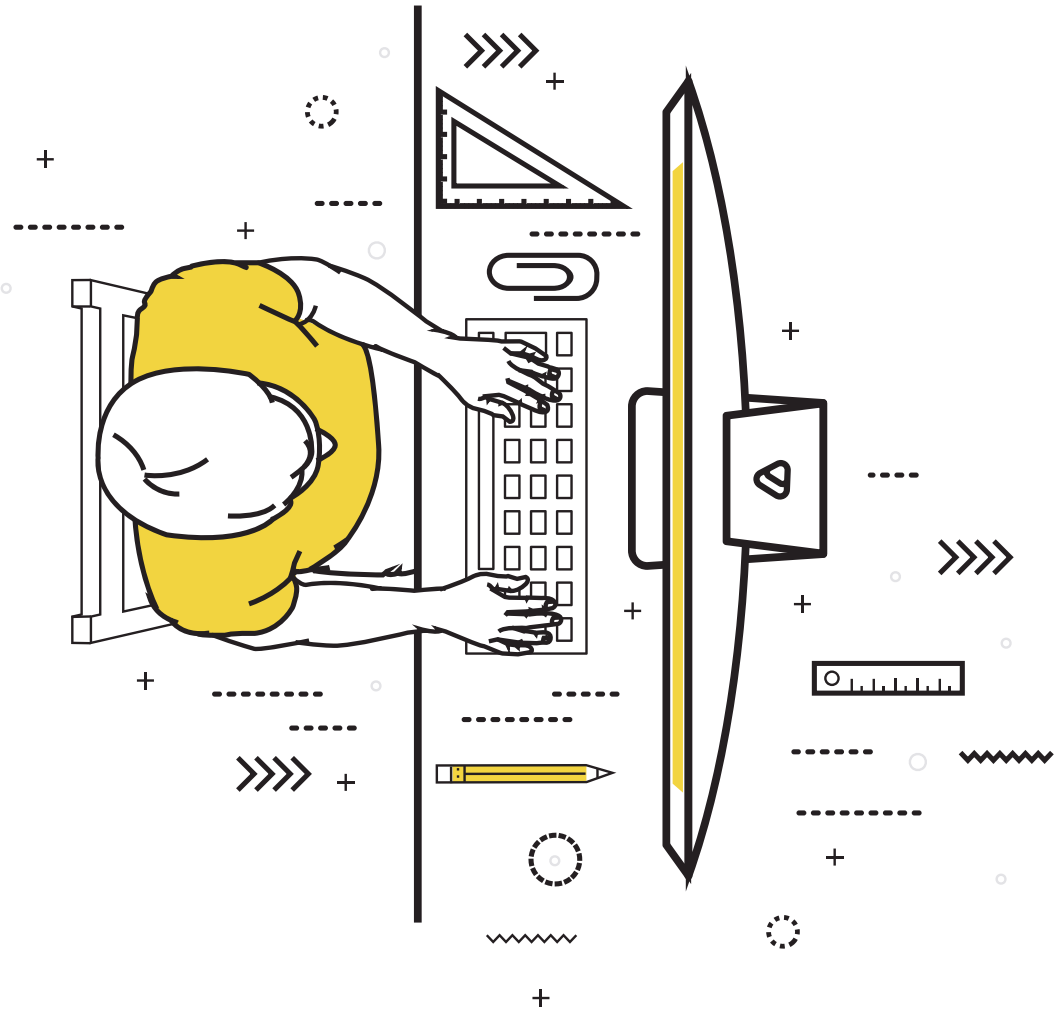
  2. **Designer API**
     The new available option needs to be communicated to the Designer back-end where the design that is being built exists. To make it possible for new components or options to be communicated to the back-end and be saved there accordingly, we extend the API's functions or add new functions for the new functionality.

  3. **Designer API**
     Applications are saved as structured designs. These designs are structured so they can be given to the Builder and be used to generate code from. Besides the back-end accepting new components via the API, it needs to be able to save these components in the new design structure. For new functionality we extend the design structure to include these new options.

  4. **Code generation**
     Triggre automatically generates code based on the designs. When new functionality is being created the code generation needs to be extended to include the new options in the application that is created with Triggre.

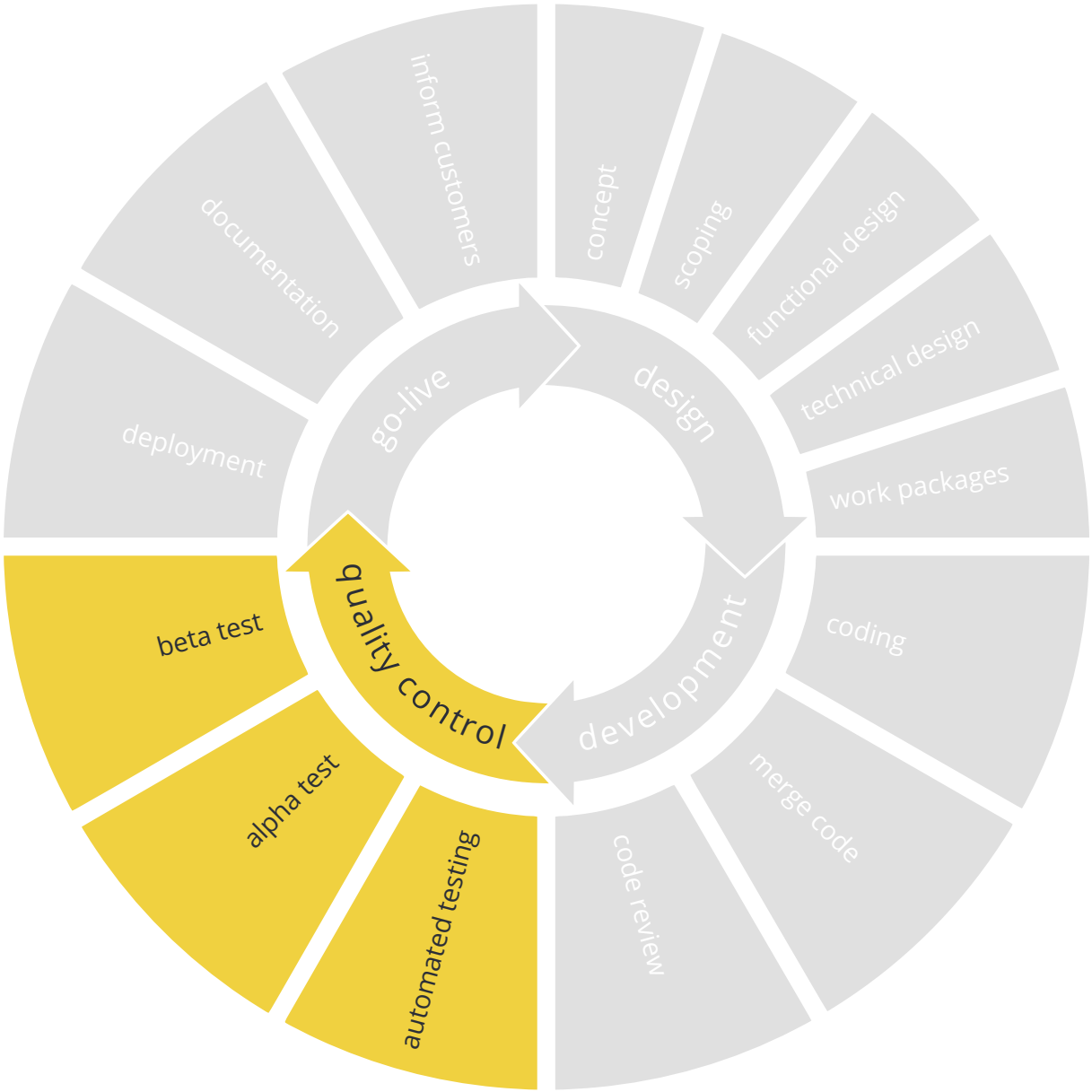
- **Merge code**
  If a feature is very broad and involves many changes, we scope what parts of the feature will be developed. Other parts are placed on the backlog for later development.

- **Code review**
  When we have scoped a feature, we start with the functional design. The functional design describes what the user is able to do, what the screens look like and how the system should behave in specific situations.



Quality control

Our quality assurance process is extremely thorough. Because the Triggre system is extremely complex in nature, our quality assurance team always has the final word whether a release is deemed ready or not.



The Quality Assurance consists of the following sub phases:

- **Automated testing**
  During the automated testing phase we run a number of test sets against our system, to make sure that there are no regressions and to test security amongst other aspects:

  1. **Functional designer tests**
     These are intended for the front-end of the Designer and test the things the user does directly.

  2. **Designer API tests**
     Even though the functional designer tests also test the API, we test the API more extensively to make sure that it does not misbehave with options the user should not be able to create in the Designer.

  3. **Unit tests**
     Besides testing the front-end and back-end of the Designer, the code of the Designer is also covered in unit tests by developers.

  4. **Code generation testing**
     Code generation is testing using both customer designs, to make sure that the applications from customers can still be created with the new functionality, as well as randomly generated designs.

     These designs of random applications are intended to cover a much wider range of possibilities than customers use, mostly to find edge cases.

5. **Security testing**
   Besides testing regular usage, we also test the following security measures:

   - **Sanitation of user supplied data**
     Not properly encoding data during transportation can lead to malicious code being injected in the application. For us this goes for the data added in the Designer, data added in the generated application, as well as making sure that we don't generate applications with malicious code added in the Designer.

   - **Generated Content Security Policy (CSP)**
     Since the user is allowed to use some limited external resources (such as pictures), we check whether the generated CSP headers are correct and allow those resources to be used in the generated application, but keep all other external content out of the application to prevent cross side scripting and cross-site request forgery.

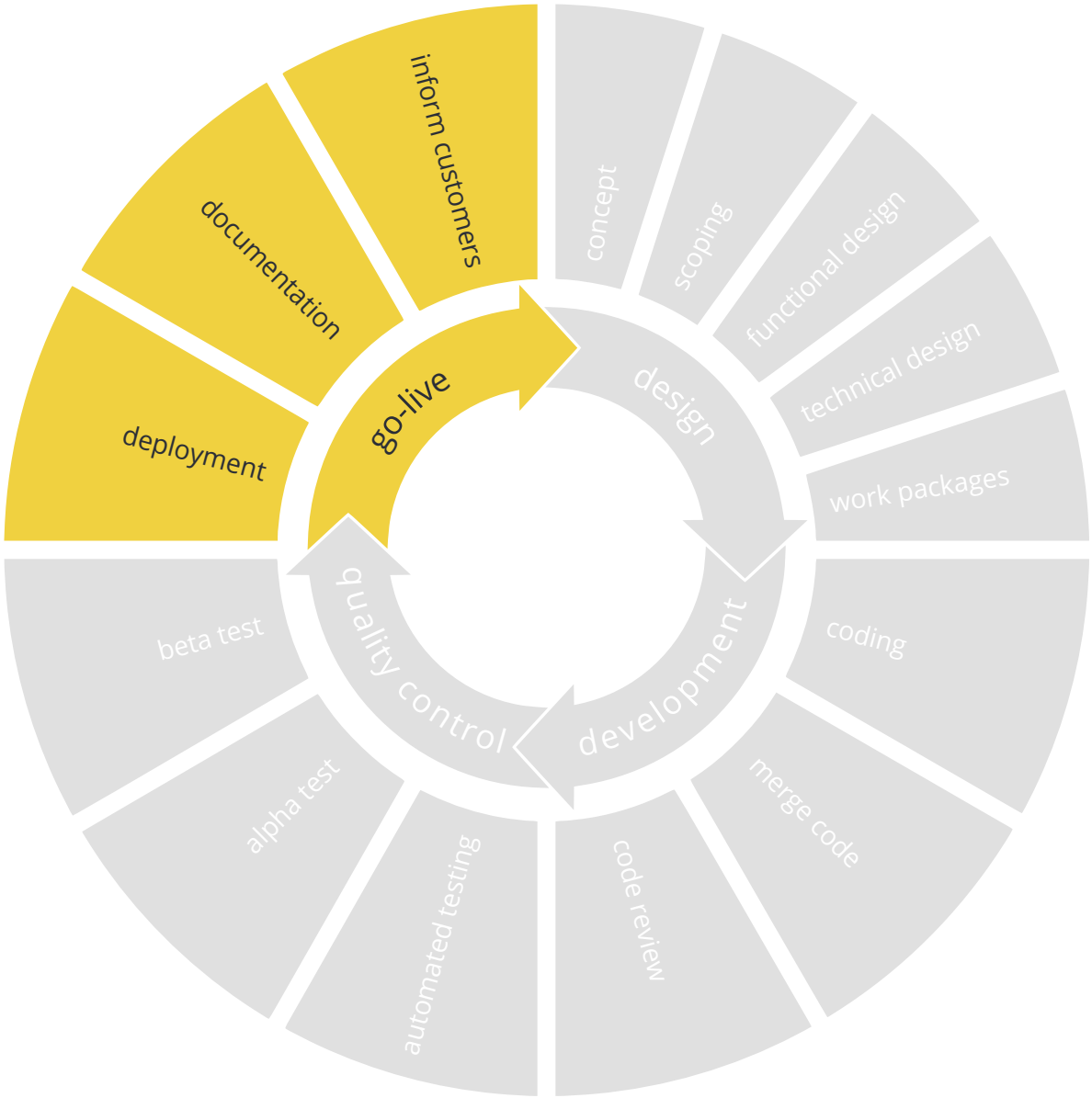
   - **Other security tests**
     These include server configuration testing related to certificate and protocol settings, but this is done in the Go-live phase.

- **Alpha testing**
  During the alpha test phase the new version is tested internally by employees at Triggre. In this phase, there is a code-freeze on new functionality. Only defects that are found are fixed in the code of the new version. When all blocking, critical and major defects are fixed, the alpha testing phase is complete.

- **Beta testing**
  In the beta testing phase, the application is tested internally at Triggre by a wider audience. During this phase, the minor and trivial defects are fixed. This phase is complete when all known defects in the new version are resolved.

Go-live

After the design, development and testing phases are completed, the new version of the Triggre platform is released during the go-live phase.



The go-live phase is where the final steps of releasing a new version of Triggre are performed:

- **Deployment**
  The new version of the Triggre platform is installed on our servers during the maintenance hours. This process is fully automated; a software package is placed in a central location, after which all servers fetch the update and install it.

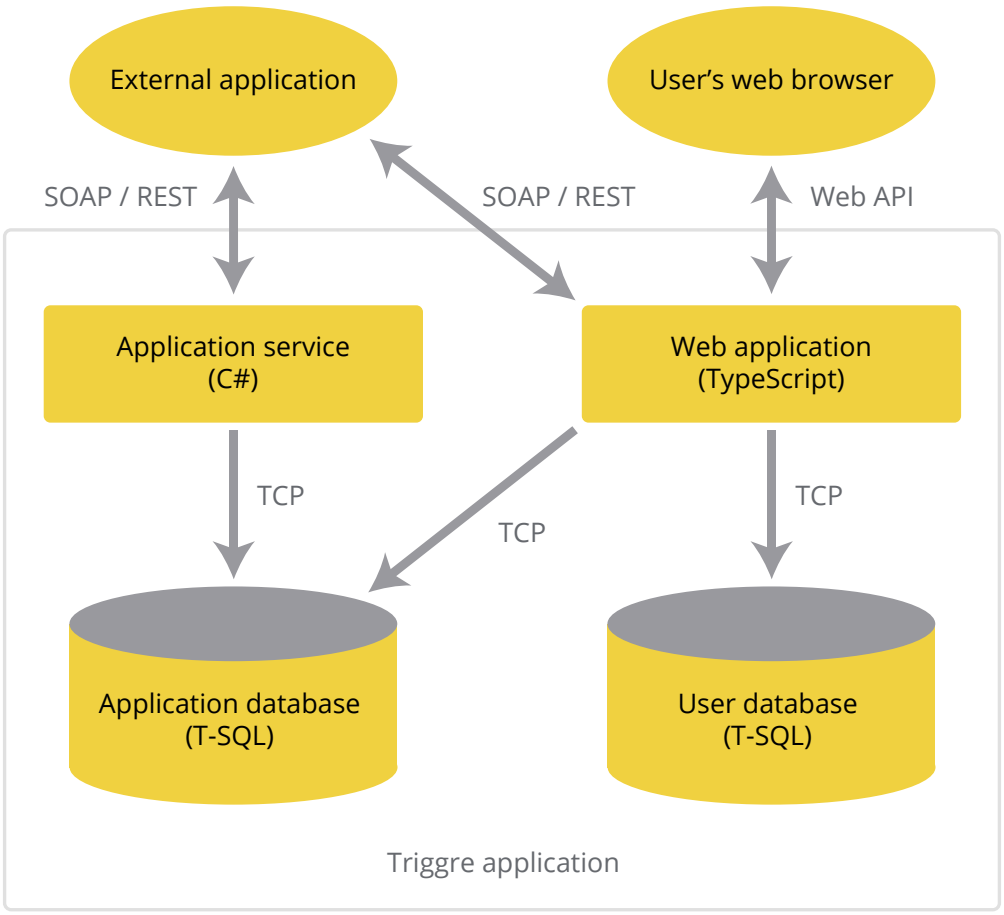
  After the installation, all customer applications are automatically re-published to make sure they get any security updates that were shipped in the new version of Triggre. This way, customers are guaranteed to have a secure application even if they haven't used the Triggre Designer in a while to publish their application.

- **Documentation**
  During the preparation of the deployment, we also update or create any internal documentation needed for the new features. This includes, but is not limited to, feature descriptions for our guides and technical documentation if so needed.

- **Inform customers**
  The last step is that Marketing informs our customers of the new release. This is done by email and by placing the information about the features in the release on our website, under release notes.

Application architecture

Triggre creates an application from a design made in the Triggre Designer. The following diagram shows the architecture of a generated application:



The architecture consists of four components: the web application, the application service, an application database and a user database.

Web application

Due to the strategic nature of the content of this document, all of the information is strictly confidential and is only available upon request. It is strictly prohibited to provide this document and/or the information held in the document to anyone without the express consent of Triggre or a representative of Triggre.

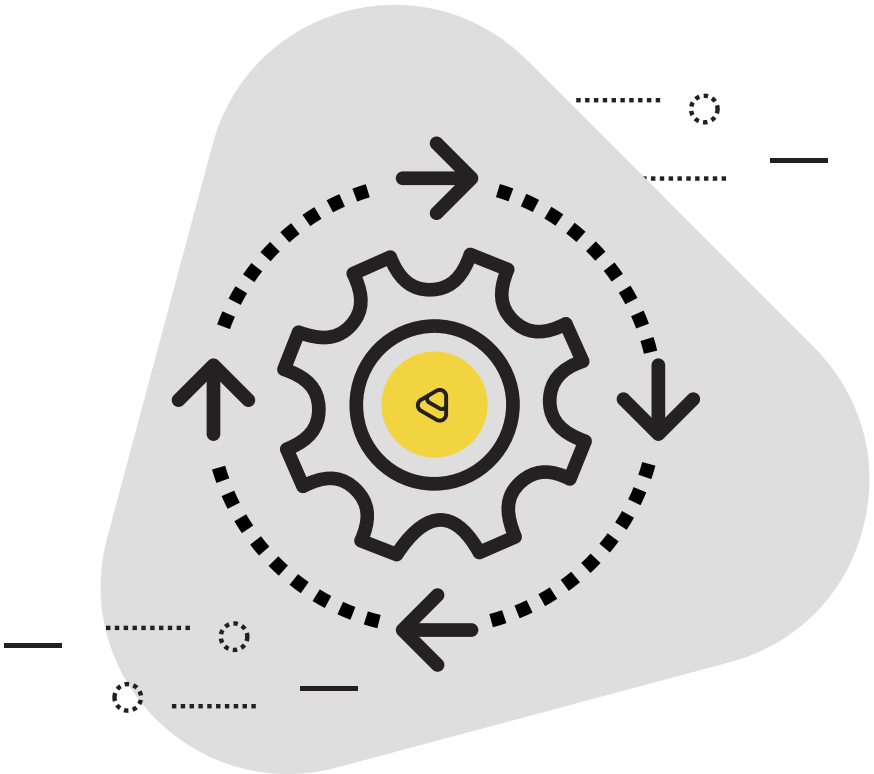
Application service

The Application service takes care of automated processes, running in the background. There are two types of automated processes: those designed by the user (called Automation flows in the Triggre Designer), and built-in processes such as the sending of queued e-mails and various clean-up activities.

Databases

Each application has two databases: the application database to store all the data the application stores and a separate database that stores user passwords (in SHA-512 hashed form with salt).

The application database can completely be encrypted to provide the maximum protection for sensitive data.



Hosting

Triggre runs completely on Microsoft Azure and is hosted in the Microsoft Western-Europe data center in Amsterdam.

Certification

The datacenters in which Triggre is hosted are amongst the best compliant in the world. Amongst these are ISO 27001, SOC 1 and SOC2. For a complete list of certifications, please check the Microsoft Azure Certification website.

Fallback location

In case of a catastrophic failure of the Western-Europe datacenter, the fallback location will immediately become active. This fallback location is the Microsoft Northern Europe datacenter, which is located in Ireland (and therefore, is still part of the European Union). This provides the best possible physical protection, while also providing strong legal protection.

Server access

Physical access to the servers is strictly reserved for Microsoft Azure engineers. Remote access to the Triggre servers is only possible from the Triggre corporate network. Connecting to the Triggre corporate network requires valid credentials, as well as a device that has a valid certificate as issued by Triggre.

The Triggre corporate network and Triggre production network are two completely separated networks, using different Access Control Lists on different domains.

Monitoring

Applications do their own monitoring and log events such as user log-ins, user actions and changes to data. Anomalies are automatically detected and reported.

Intrusion and anomaly detection

Unexpected behavior in the application is automatically detected and reported by the Triggre Platform. These reports are evaluated and based on that evaluation appropriate actions are taken immediately.

All application processes in Triggre have an expected way of executing them (as specified in the Designer). Any deviation from the expected execution (such as errors) cannot be executed because of this and is automatically reported by the Triggre Platform. This also, for example, reports URL manipulation (or URL injection) and the loading of external resources.

Code injections

Other attempts at code injections are not reported, but instead are sanitized so the code isn't executed. Because of the nature of Triggre, where you can create a wide range of applications, saving a bit of text that looks like a script might be valid for certain applications. So instead of forbidding that or raising alarms, Triggre properly encodes during transport so the code is never executed within the application.

Platform monitoring

On a platform level, monitoring is done by Windows' built-in monitoring, extended with Triggre's own resource monitoring system. It tracks events on a platform level, such as user log ins, whether processes are running as they should and it reports when resource consumption get to high, which might negatively influence application performance.

Event monitoring

Currently event monitoring is decentralized. An implementation of ELK stack for the monitoring of events is currently being researched.