# SECURITY PRACTICES

**CONFIDENTIAL - SECURITY-SENSITIVE INFORMATION**

**Last update: November 21st, 2019**

# DISCLAIMER

*This document describes the Newired Security Practices. Information provided within this document is confidential and only available for the intended recipient and may not be used, published or redistributed without the prior written consent of Newired. Disclosure of any part of this document to other parties is subject to penalties.*

***The parties undertake to sign the NDA annex that specifically regulates the mutual duties of confidentiality.***

# Vision

At Newired we deliver 100% code-free tools that allow the delivery of quick usability fixes, speed up onboarding on any web application and put users at the center of software delivery and development.

Our success comes from the success of our customers. We constantly develop solutions that must have a positive impact on their daily life by speeding up their Web experience and making it more pleasant. Anything that could be represent a hazard for them is seriously considered and pursued. As a modern reality, Newired is oriented to S-SDLC methodologies to offer an agile software lifecycle management and guarantee quality, speed and an adaptive approach to software development.

Newired adheres to the OWASP Top 10 and OWASP SAMM recommendations.

# Solutions

## Newired Portal

Newired Portal is the online portal that hosts your Newired Sites, and stores the data for the Journeys you create for them using the Newired Editor app.

Portal uses PostgreSQL database to store all data.

## Newired Editor

Newired Editor is the Journey's builder application, it requires the Newired Portal to store the created Elements.

## Newired Launcher

Newired Launcher is the Newired component that contains all Journeys created with Newired Editor and Newired Portal. Once deployed in the Underlying Application, it loads the Journeys to be visible by End Users.

## Newired Snippet

Newired Snippet is a small piece of JavaScript code that must be inserted into the Underlying Application. The role of a Snippet is to download the Newired Launcher from the Delivery Server and deploy it into the Underlying Application.

## Delivery Server

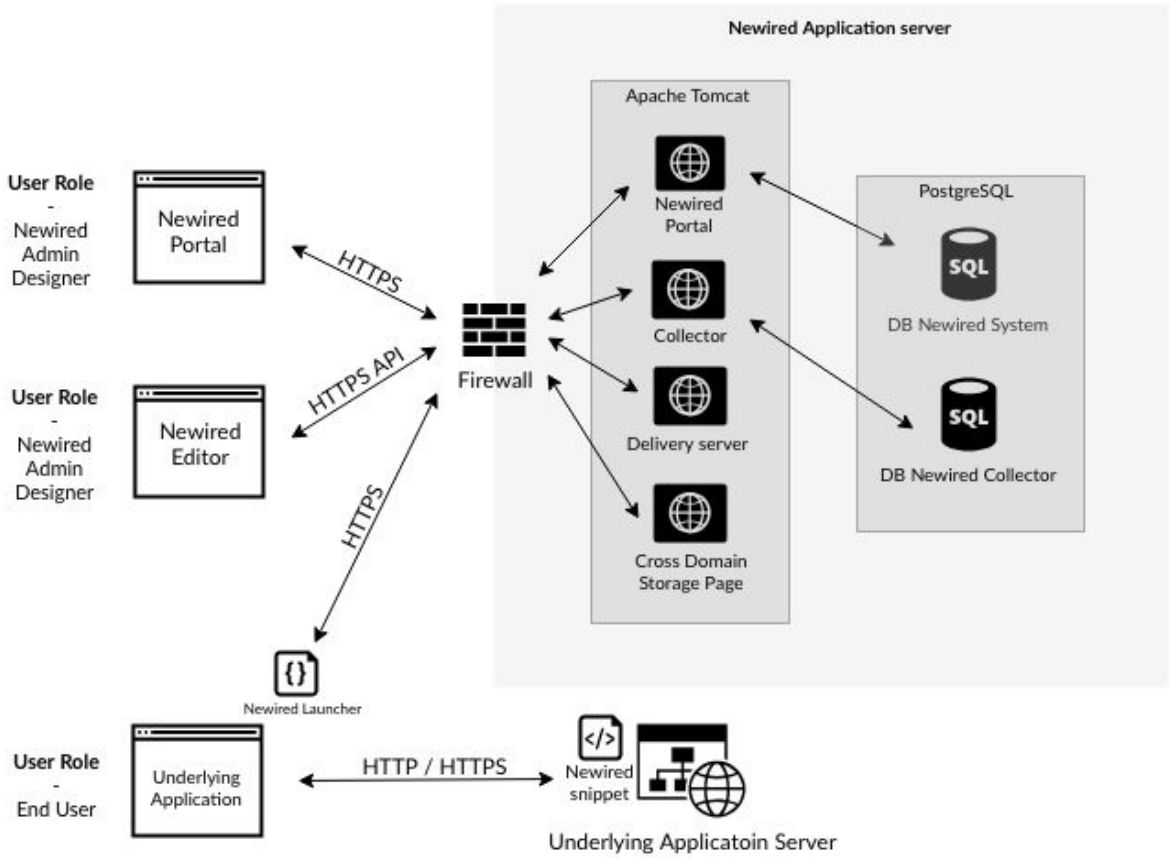
Delivery Server provides Journeys to Newired Launcher deployed to Underlying Application.

## Cross-Domain Storage Page

Pages which provide access to a cross-domain storage need to run Journeys across multiple web domains.

## Newired Collector

Newired Collector is the online service to collect user's feedback and reports. Feedbacks and reports are sent to Collector by  Launcher deployed within Underlying Application.



**Newired Application Server** consists of

- Apache Tomcat
- PostgreSQL database

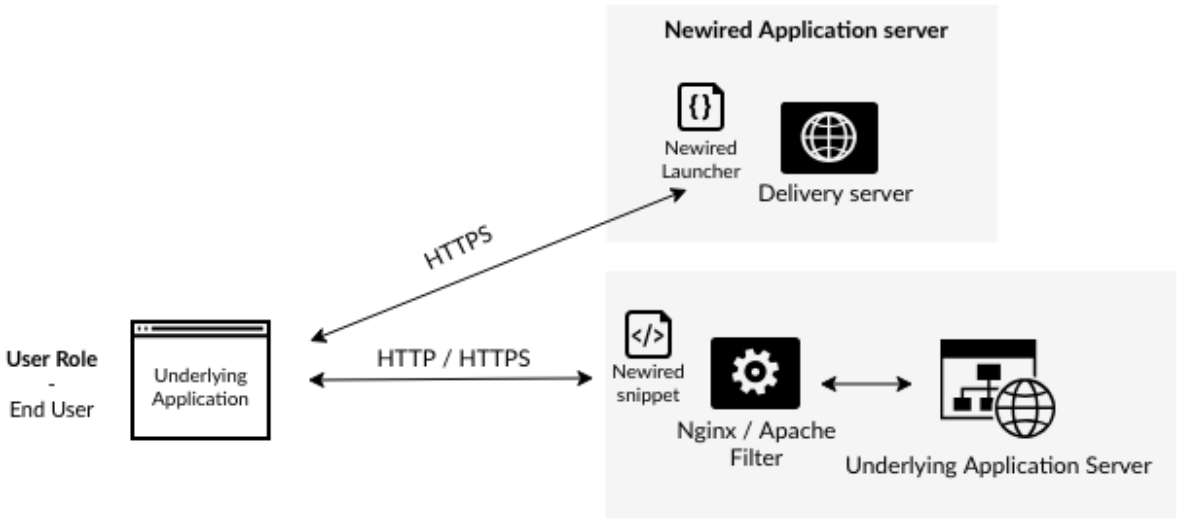
Apache Tomcat hosts Newired Portal, Newired Collector, Cross-Domain Storage Page and Delivery Server. These components can be operated in several configurations outside Newired Application Server.

## Scenarios

### Newired Snippet

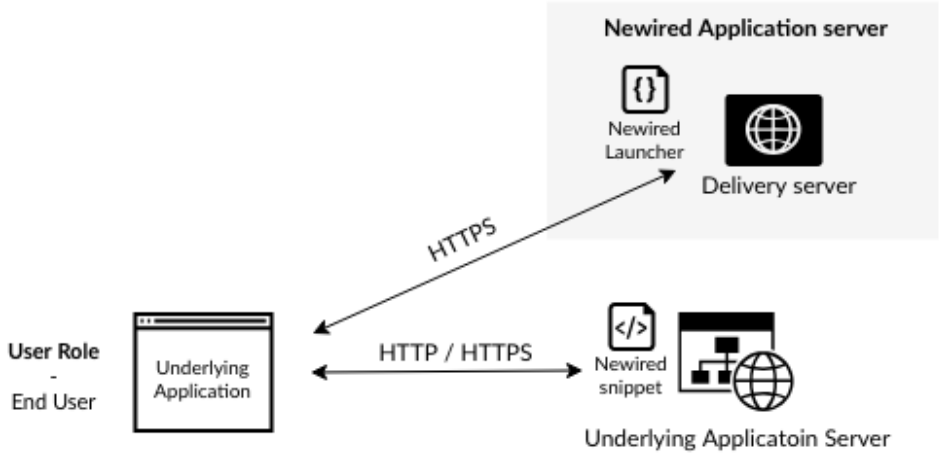
Newired Snippet is meant to be presented inside any page of Underlying Application where Newired Journeys should be available. This can be achieved by two methods.

1. Insert Snippet directly into Underlying Application HTML pages.
2. Configure HTTP filter on Web server that hosts Underlying Application.

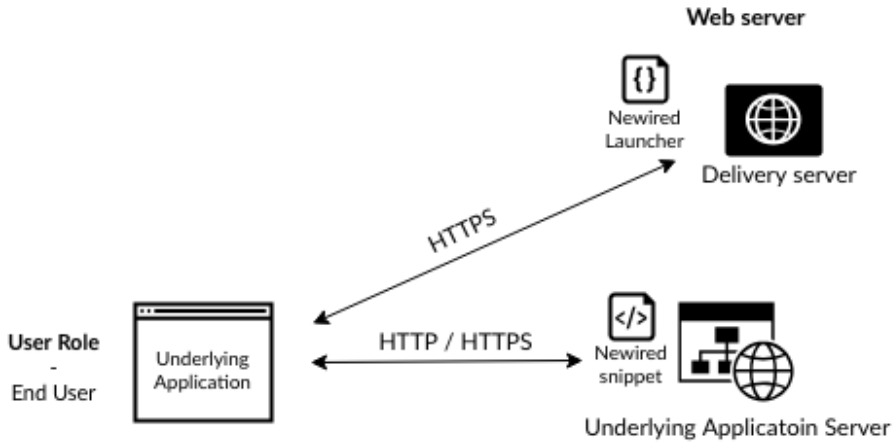



### Delivery Server

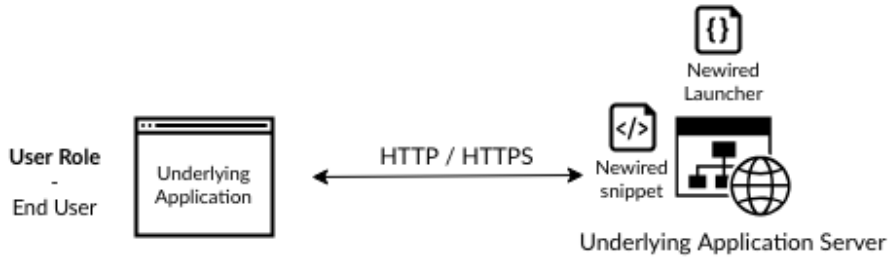
Delivery Server is technically a set of JavaScript, CSS and JSON files accessible through HTTPS. A Newired Snippet downloads all files from the the Delivery Server. The easiest way is to use Delivery Server provided by Newired Application Server.




Anyhow, users can decide to put those files on any alternative Web server. A typical motivation for this step is to avoid the dependency from Underlying Application on Newired Application Server.




Alternatively, they can put those files into Underlying Application Server.

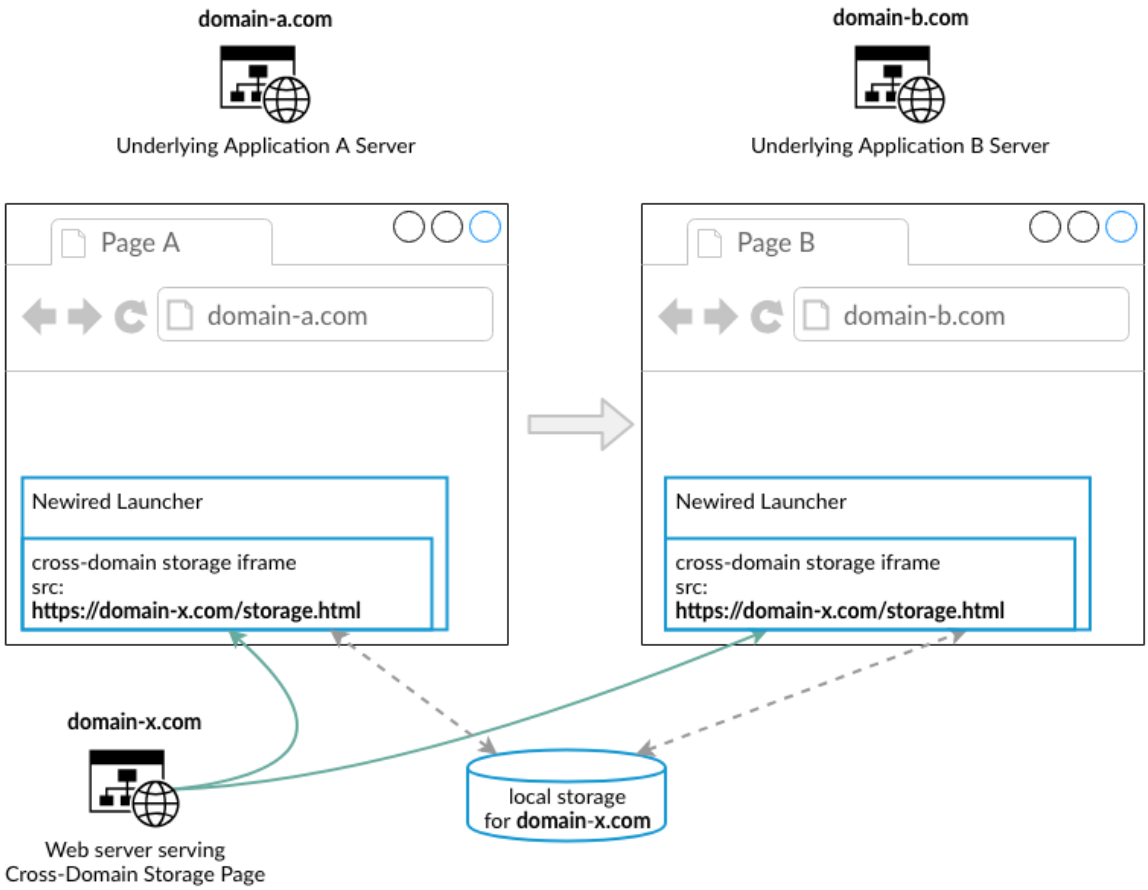



Collector

Whenever customers want to collect reports about end user's interactions with Journeys or to collect user's feedback then Newired Launcher has to have access to Newired Collector service. First option is to run Newired Collector inside Newired Application Server. Second option is to run Newired Collector in a separate Application Server (Apache Tomcat). A typical motivation for this step is to avoid dependency of Underlying Application on Newired Application Server.

Cross-Domain Storage Page

Cross-Domain Storage Page is a technique that allows to run one Journey across multiple domains (multiple Underlying Application running on different web domains). This page has to be accessible from end-user's computers. For this reason there are two possibilities where to have the Cross-Domain Storage Page. By default, it is provided by Newired Application Server but there is the option to put this page to any alternative Web server.




Browser's local-storage for domain *x.com* serves as cross-domain storage for Journeys going through Underlying Applications A and B.

## Industries

*Newired is currently delivered to companies belonging to several kinds of industry: Utilities, Finance, Mechatronics, Telecom and Banking among the others. Since the Newired's mission is to cover the gap between applications and users by bettering the UX, our solution is suitable for any industry leveraging on web-technologies to increase the business. The fast growth of the variety of companies that is adopting Newired is a proof of the flexibility and effectiveness of this software as it horizontally joins needs of users approaching web applications regardless of the business case Newired is going to empower.*

# Security Assurance Program

*As a software company, Newired releases its solution to customers as a company-wide on-boarding and training framework. Our Security Assurance Program focuses on customer's security by empowering the process of security testing, code review and S-SDLC practices. This is the best mean for a secure release process and matches Construction and Verification security objectives of the OWASP SAMM Framework. We continuously collect feedback from customers and keep our teams up to date with security practices. From one side we care about operational enablement prioritizing customer's security issues through appropriate communication channels in a process of Security and Threat Issue Management for our Deployment objectives. On the other side, we have started a path of internal periodical trainings to keep Newired at a nice level of knowledge about defensive programming, security by design practices and we are embodying those habits in our Governance objectives.*

*Newired knows where to put extra effort and we are evaluating routes to improve our internal infrastructure to improve business continuity capabilities. We have assessed our internal assets and planned next steps to be included within Deployment and Governance objectives as future actions.*

*Nothing in Newired is static. We continuously review our roadmap and try to tune security practices and fix issues. We have started from a very embryonal security management and we have got in a few years to a clear path to growth internally and with our customers adopting practices of internal auditing to ensure compliance with our security processes and practices.*

# Security Assurance Assessment

By measuring our Newired organization against the defined Security Practices, an overall picture of built-in security assurance activities is created. This type of assessment is useful for understanding the breadth of security activities currently in place at Newired organization. Further, it enables us then to utilize SAMM to create a future roadmap for iterative improvements.

## *Governance*

### Strategy & Metrics

- *SM1 - Is there a software security assurance program already in place?*
  The Assurance program is documented and accessible to Newired personnel. It has been used in recent development efforts. The team members receive training against assurance program and responsibilities.
- *SM1 - Is most of your development staff aware of future plans for the assurance program?*
  The Assurance program goals are shared with relevant team members. Assurance program goals have been presented to key developers. A roadmap has been put in place to reach those goals.

- *SM1 - Do most of the business stakeholders understand your organization's risk profile?*
  The Newired organization has documented the motivation behind creating a software security assurance program. Assurance program has been customized to align with the organization's motivation and goals. Worst-case scenarios for Newired application and data assets have been collected and documented. Scenarios, contributing factors, and mitigating factors have been reviewed with business owners and other stakeholders.

- *SM2 - Are most of your applications and resources categorized by risk?*
  Most critical components, Newired Snippet and Newireed Launcher, which are injected into the underlying application, are prioritized.

## Policy & Compliance

- *PC2 - Does the organization utillize a set of policies and standards to control software development?*
  A set of security policies has been created. Requirements based on known business drivers for security have been added to security policies. The policies are mostly derived from the OWASP Top 10 Project. Security policies do not include requirements that are too costly or difficult for project teams to comply with.

- *PC2- Are project teams able to request an audit for compliance with policies and standards?*
  Activities already performed in the area of software security let the RnD team request security audits. Internal reviews are prioritized based on business risks involved. Review results are analyzed by project stakeholders.

## Education & Guidance

- *EG1 - Have most developers been given high-level security awareness training?*
  Application security awareness training is provided to key developers. Training covers topics such as common vulnerabilities and best practice recommendations for eliminating vulnerabilities. The development team is familiar with the OWASP standards for common threats,

possible attacks, and remediations.

- *EG1 - Does each project team have access to secure development best practices and guidance?*
  Resources regarding secure development practices have been assembled and made available to developers. Management informs developers that they are expected to utilize secure development resources. The developers access the security practices derived from the OWASP framework and adjusted for specifics of the Newired product. Sharing is done using the Polarion ALM document management capabilities.

# Construction

## Threat Assessment

- *TA1 - Do most projects in your organization consider and document likely threats?*
  The developers have analyzed and documented possible security threats. The review and adjustment sessions are scheduled quarterly to keep the list up to date. Items on the list include both industry standards and specifics of the Newired technology.

- *TA3 - Do project teams specifically consider risk from external software?*
  Third-party, external libraries and code used in each project are clearly identified and documented for each project. The review is done quarterly.

## Security Requirements

- *SR1 - Do most project teams specify some security requirements during development?*
  There are two types of security requirements a) non-functional security requirements that apply across all the features (e.g. all private data has to be stored encrypted) - these are documented and checked as a part of “definition of done” for every User Story. And b) functional security requirements such (e.g. ask for an SSL certificate during installation procedure). Such requirements are created during Epic decomposition and are tracked as User Stories.

- *SR1 - Do project teams pull requirements from best practices and compliance guidance?*
  Industry best practices are used to derive additional security requirements to further improve the security aspect of the product. We use OWASP as a reference list of practices.

- *SR2 - Are project teams specifying requirements based on feedback from other security activities?*
  When identified, additional security requirements are created based on feedback from code reviews, penetration tests, or other security activities.

## Secure Architecture

- *SA1 - Are project teams provided with a list of recommended third-party components?*
  A list of commonly used third-party libraries and code is collected and known among key developers. The libraries are informally evaluated for security based on past incidents, responses to identified issues, complexity, and appropriateness to the organization. A list of approved third-party libraries for use within development is maintained by the Chief Architect.

- *SA1 - Are most project teams aware of secure design principles and applying them?*
  The design principles based on the OWASP framework are shared with key developers.

# Verification

## Code Review

Code review practices are part of the Newired development process since the very beginning.

- *CR1 - Do most project teams have review checklists based on common problems?*
  The organization has derived a short lightweight code review checklist based on previously identified security requirements.

- *CR1 - Are project teams generally performing review of selected high-risk code?*
  High-risk software components are prioritized above other code during the review process. Remediation of findings in high-risk components are prioritized appropriately.

- *CR2 - Can most project teams access automated code analysis tools to find security problems?*
  Automated static code analysis tools have been integrated within the development process and are executed on regular bases. The developers have access to the results.

- *CR2 - Do most stakeholders consistently require and review results from code reviews?*
  Project stakeholders review and accept any risks that were chosen not to address. Project stakeholders are creating a plan for addressing findings in legacy code if needed.

- *CR3 - Do project teams utilize automation to check code against application-specific coding standards?*
  The organization uses automated tools that are run daily or ad-hoc upon request. There are no application-specific settings at this moment.

## Security Testing

- *ST1 - Are projects specifying some security tests based on requirements?*
  The team identified basic sub-set of automated tests described by OWASP ZAP. Those tests are

automatically executed and results shared with both development and quality assurance people.

- *ST1 - Do most projects perform penetration tests prior to release?*
  The organization performs basic sub-set of penetration tests. Penetration testing issues are resolved to an acceptable level of risk prior to release.

- *ST1 - Are most stakeholders aware of the security test status prior to release?*
  Penetration testing issues are reviewed with project stakeholders, mainly with Product Manager and CTO. They select issues to remediate prior to release.

- *ST2 - Are projects using automation to evaluate security test cases?*
  The organization chose and uses several tools for automated security testing. From those tools, the organization chose a basic subset of tests. Automated security testing has been integrated within the development process.

- *ST2 - Do most projects follow a consistent process to evaluate and report on security tests to stakeholders?*
  The Newired organisation evaluates results of security tests in the development team. There is a plan of improvements to ensure better sharing of this information with other stakeholders, which will be necessary in relation to the organizational growth.

# *Deployment*

## Vulnerability Management

- *VM1 - Do most projects have a point of contact for security issues?*
  In the early phase the role of Customer Care Manager operates also a single point of contact for this type of issues.
  Customer support at support@newired.com takes this responsible for all live customers.

- *VM2 - Does the organization utilize a consistent process for incident reporting and handling?*
  Customer contacts Newired support via email. Issue is tracked in Polarion ALM and planned in Kanban board if reproducible. If not reproducible, it is closed as “Not a bug” in Polarion.

## Environment Hardening

- *EH1 - Do the majority of projects document some requirements for the operational environment?*
  The organization documents and maintains a set of baseline operating platforms. The key developers discuss  assumptions made about operating environments during development.

Organization and project operating platforms are reviewed at least every six months.

- *EH1 - Do most projects check for security updates to third-party software components?*
  The development organization regularly monitors software components for security updates. Critical software component updates are applied once identified.

## *Main Scenarios*

### Secure Newired Application Server

Newired Application Server  is secured in several ways.

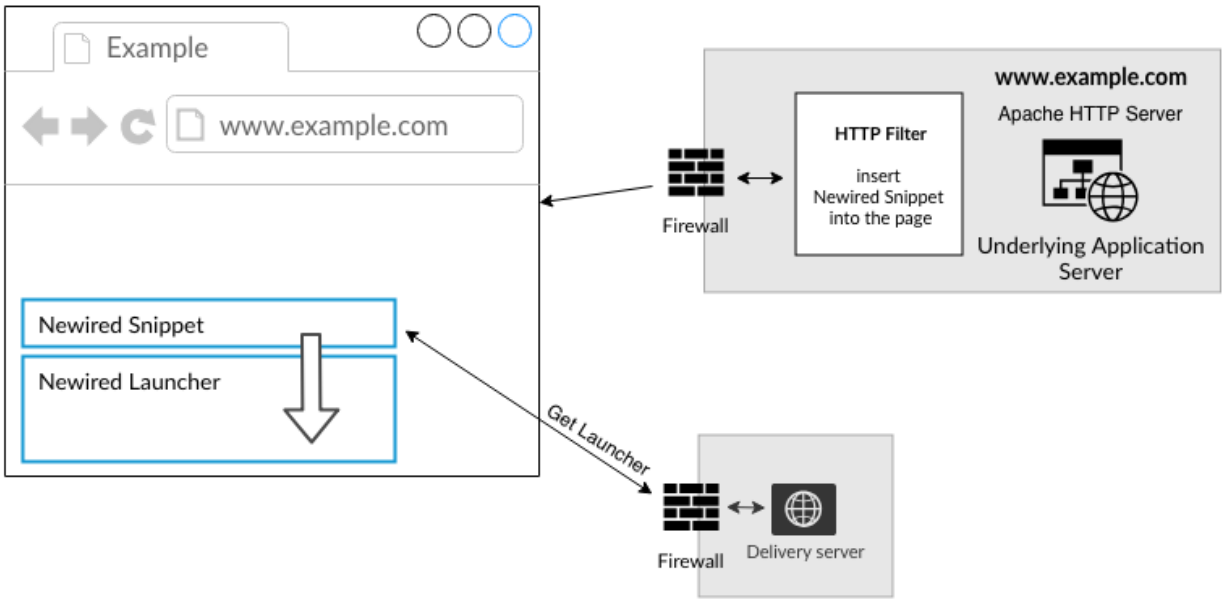
- All connections are configured to use HTTPS.
    - Newired Application Server Hardening document describes how to configure Apache Tomcat server to use HTTPS.

Future steps to improve Newired Application Server security

- Installed Apache Tomcat will run under dedicated account with limited privileges.
- PostgreSQL database will be created by Newired Installer using dedicated user with limited privileges.

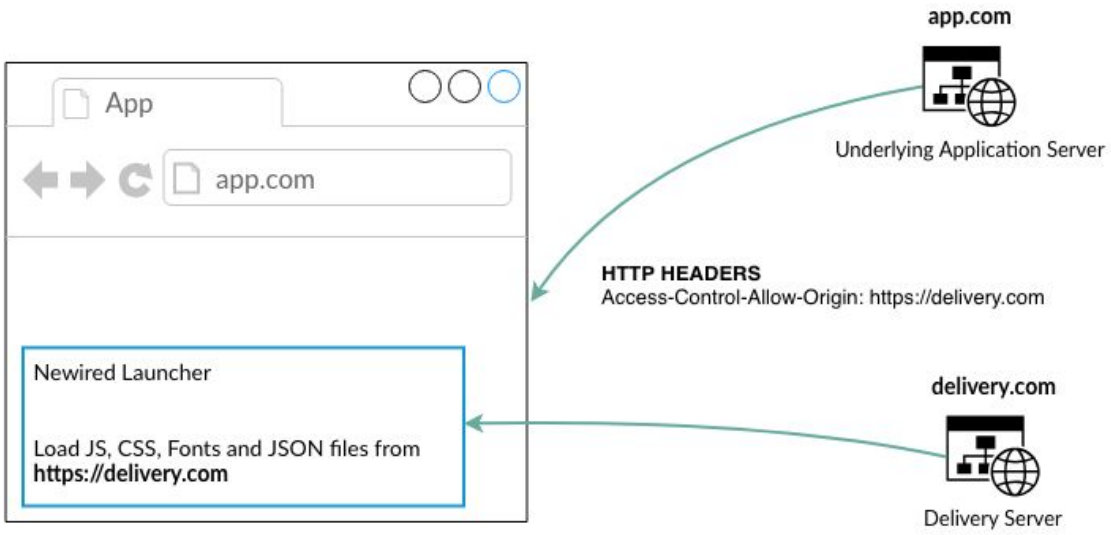
### How Snippet can be inserted into page

When Newired Snippet is inserted into Underlying Application page by HTTP filter on Underlying Application Server then security recommendation described in Newired Application Server Hardening document has to be followed.



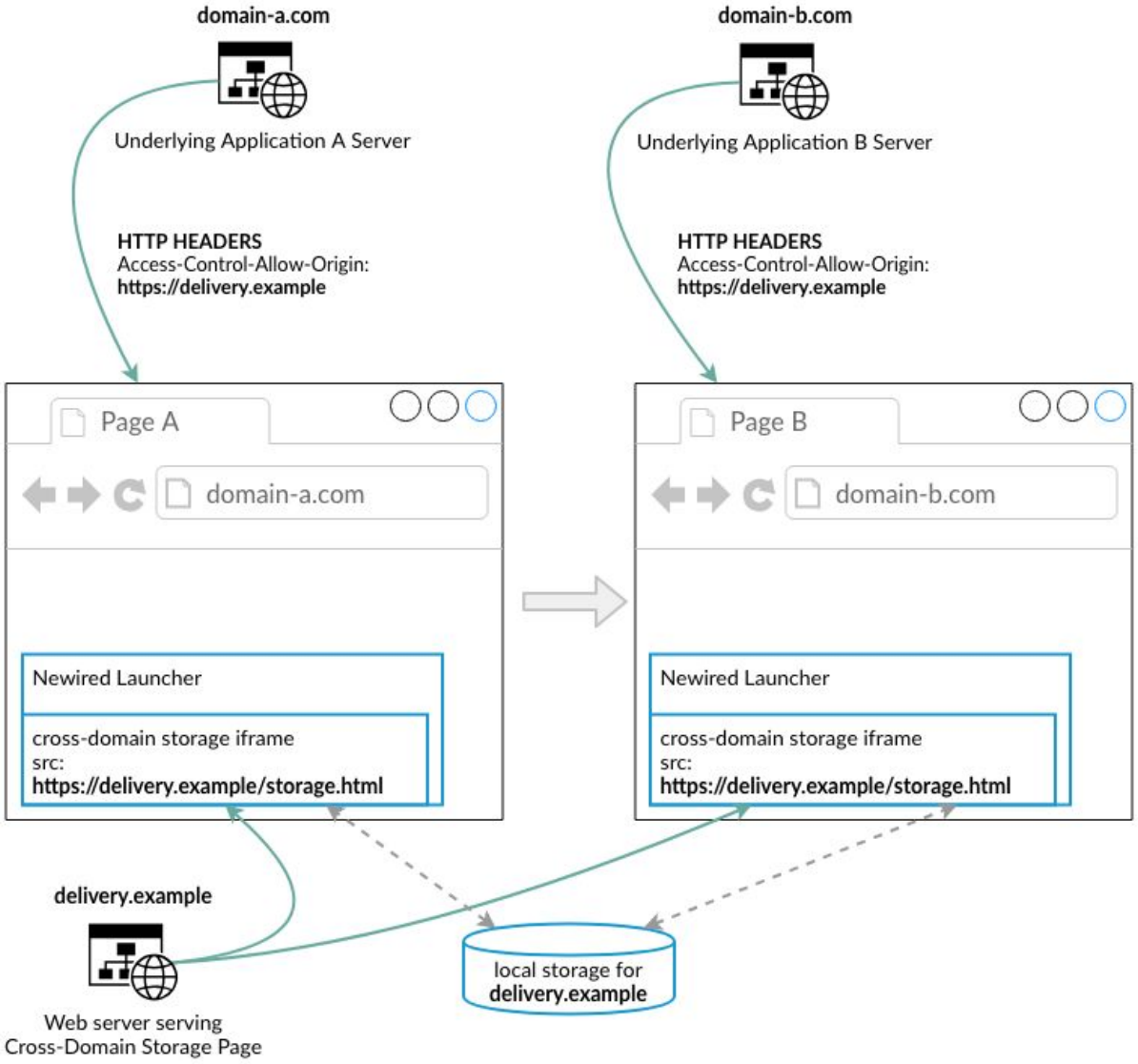

Newired Launcher running inside Underlying Application is using several additional resources. These resources are downloaded from Newired Delivery Server that can be provided by Newired Application itself of by custom Web Server.

In all cases is needed to configure CORS HTTP headers properly to allow download all resources but not compromise security of Underlying Application.

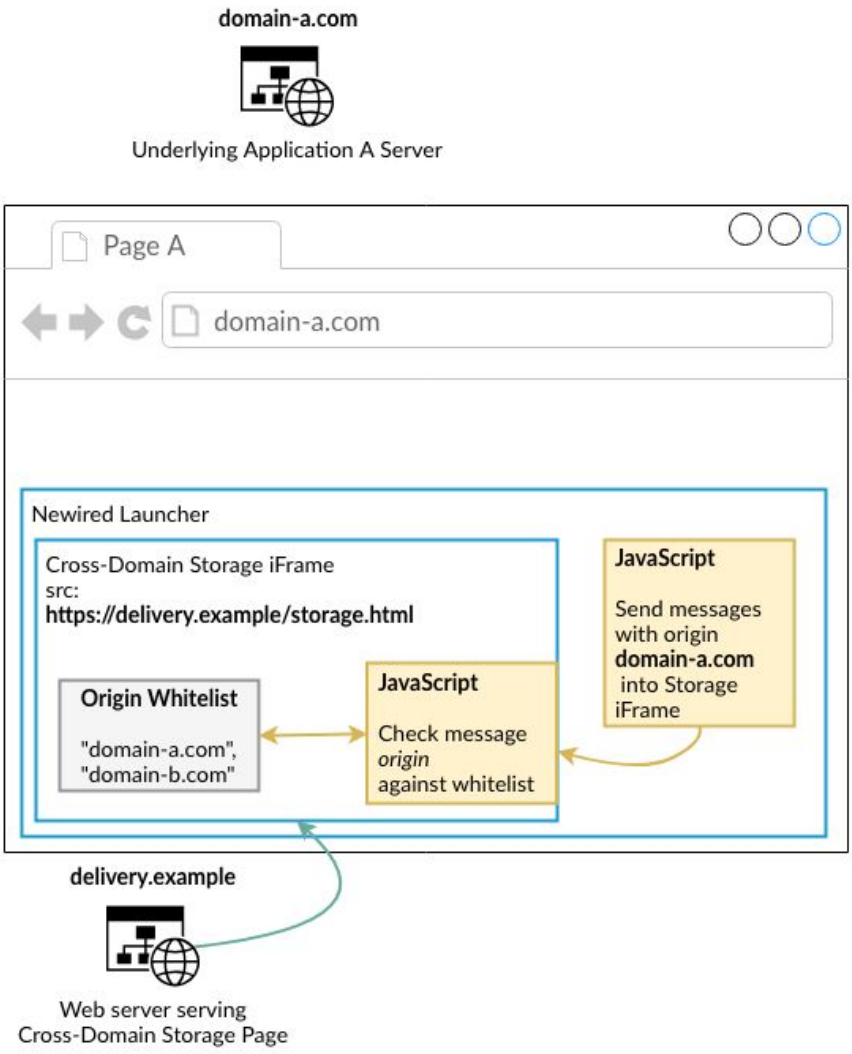



## Cross-Domain Journeys

In order to use cross-domains Journeys is needed to configure properly Cross-Domain Storage Page. To not compromise security of Underlying Application is needed proper configuration of CORS HTTP headers on Underlying Application Server and Delivery Server that provides Cross-Domain Storage Page.

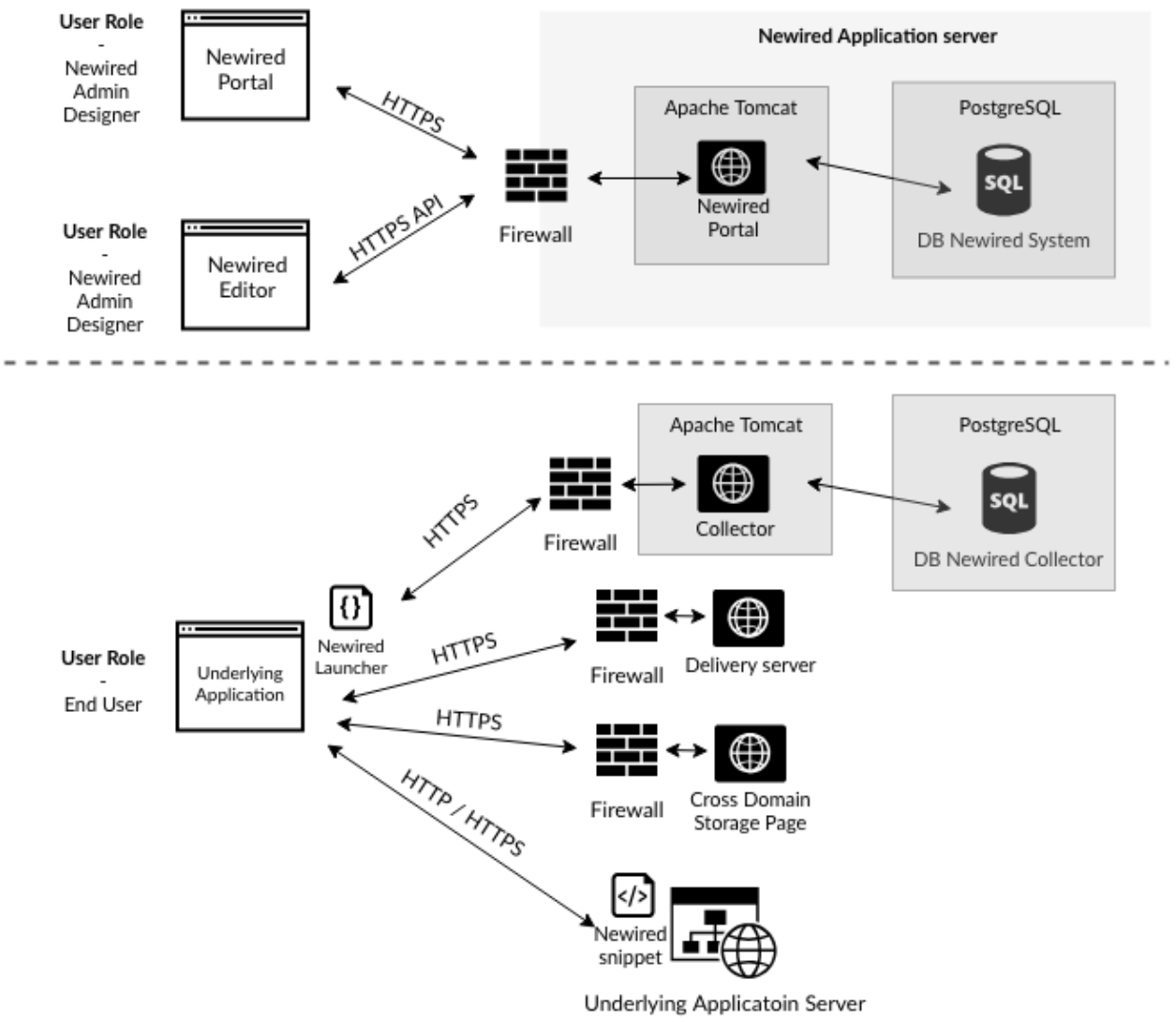



Cross-Domain Storage Page is loaded into *iframe* created in Underlying Application page and JavaScript inside this *iframe* communicate with JavaScript inside Underlying Application page. This communication is secured by white-listing of messages incoming into iframe.




## How to isolate Underlying Application from Newired Application Server

ll services Newired Launcher needs do not require connectivity to the Newired Application Server.  It is possible to run the Underlying Application with Newired Launcher without connectivity to Newired Application Server (End users can access the Underlying Application Pages but do not need access to Newired Application Server).




Steps to separate services from Newired Application Server

- Establish Delivery Server outside Newired Application Server.
    - Journeys (published from Newired Portal as ZIP package) will be provided via HTTPS from this Delivery Server.
- Establish Cross-Domain Storage Page outside Newired Application Server.
- All details how to configure Delivery Server and Cross-Domain Storage Page securely is described in Newired Application Server Hardening document.

## *Threat Assessments*

Newired uses OWASP recommendations to address the main areas of threats.

## A1:2017 Injection

*Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.*

All requests (SQL) passing to DB are done by layer which avoid injecting risks (Hibernate).

## A2:2017 Broken Authentication

*Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.*

## A3:2017 Sensitive Data Exposure

*Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.*

The whole REST API is covered by authorization check in order to deny any unauthorized access.

## A4:2017 XML External Entities (XXE)

*Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.*

## A5:2017 Broken Access Control

*Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.*

Any request into backend (REST API) is subjected to authorization process.

## A6:2017 Security Misconfiguration

*Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched and upgraded in a timely fashion.*

Newired Application Server components (like Apache Tomcat) are installed with secure configuration. Other security recommendations related to the system configuration are described in the Newired Application Server Hardening document. To get a secure setup, they must be followed.

3rd party components are subjected to audit to avoid risks caused by using outdated versions.

## A7:2017 Cross-Site Scripting (XSS)

*XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.*

All output generated into page (HTML) is safely escaped by framework used for all Newired UI (Elm)

## A8:2017 Insecure Deserialization

*Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.*

## A9:2017 Using Components with Known Vulnerabilities

*Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.*

3rd party components are subject to security audit that checks known vulnerabilities and outdated state.

## A10:2017 Insufficient Logging & Monitoring

Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.

## Injection of Newired Journeys in the Underlying Application

Newired takes care about the quality of Newired components injected into the Underlying Application page. This effort can be divided into three main areas.

Functionality

In order to avoid any unwanted influence of Newired components to Underlying Application Newired follows these rules

- All JavaScript objects, functions and 3rd party entities are loaded in way that avoid conflicts with objects, functions and 3rd party entities used originally in Underlying Application.
- All JavaScript code of Newired components is checked by static code analysis.
- Newired Snippet with all subsequent components are loaded into the page as last thing.

Security

- Newired components are subjected to penetration tests.
- All JavaScript code of Newired components is checked by static code analysis.
- All 3d party entities loaded into Underlying Application page are audited for know security issues.

Performance

- Newired components are subjected to performance tests.

# S-SDLC and Security Practices

Newired releases quarterly when each release is managed utilising Scrumban methodology focusing on quality in each Kanban phase with final certification at the end of each release.

All work is managed using a software tool Polarion ALM to ensure traceability and process guidance.

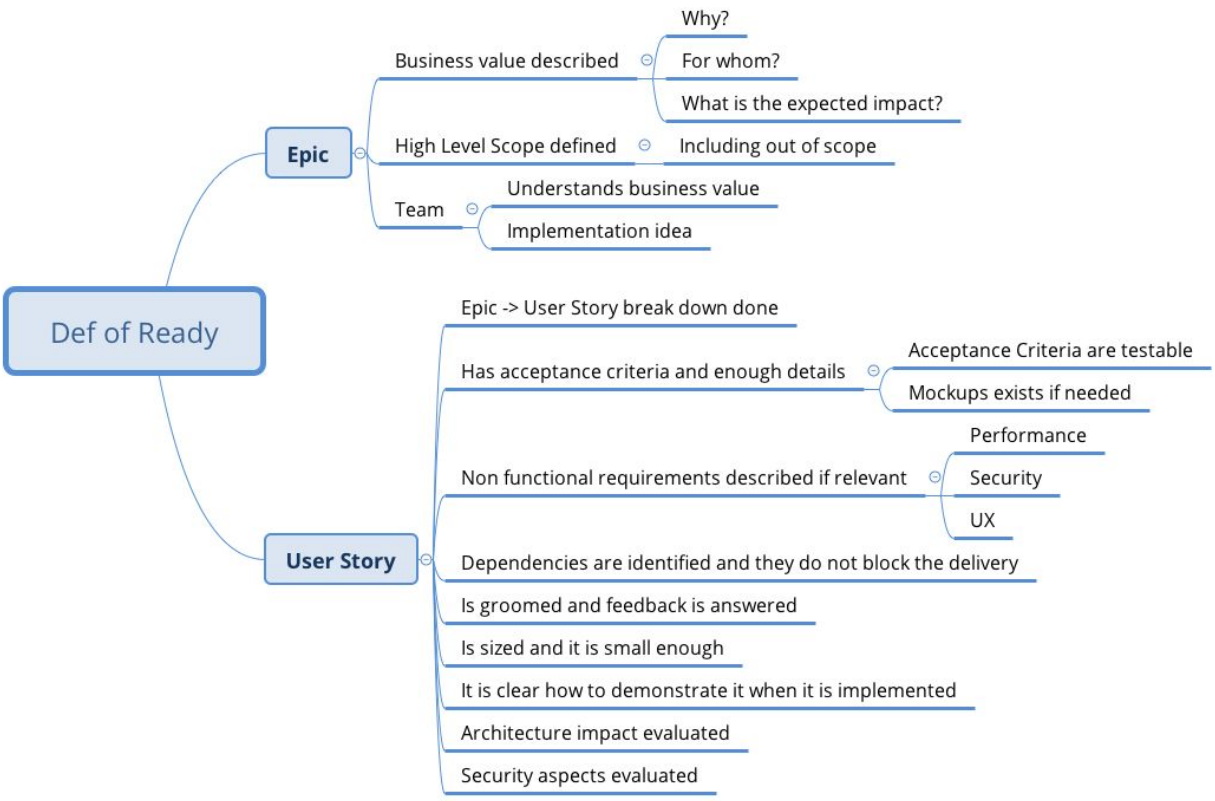
## *Definition of Ready and Done*

These definitions are team agreements on the activities and criteria to be done for every work item.

Both of these definitions are living documents and they are evaluated during regular retrospectives.

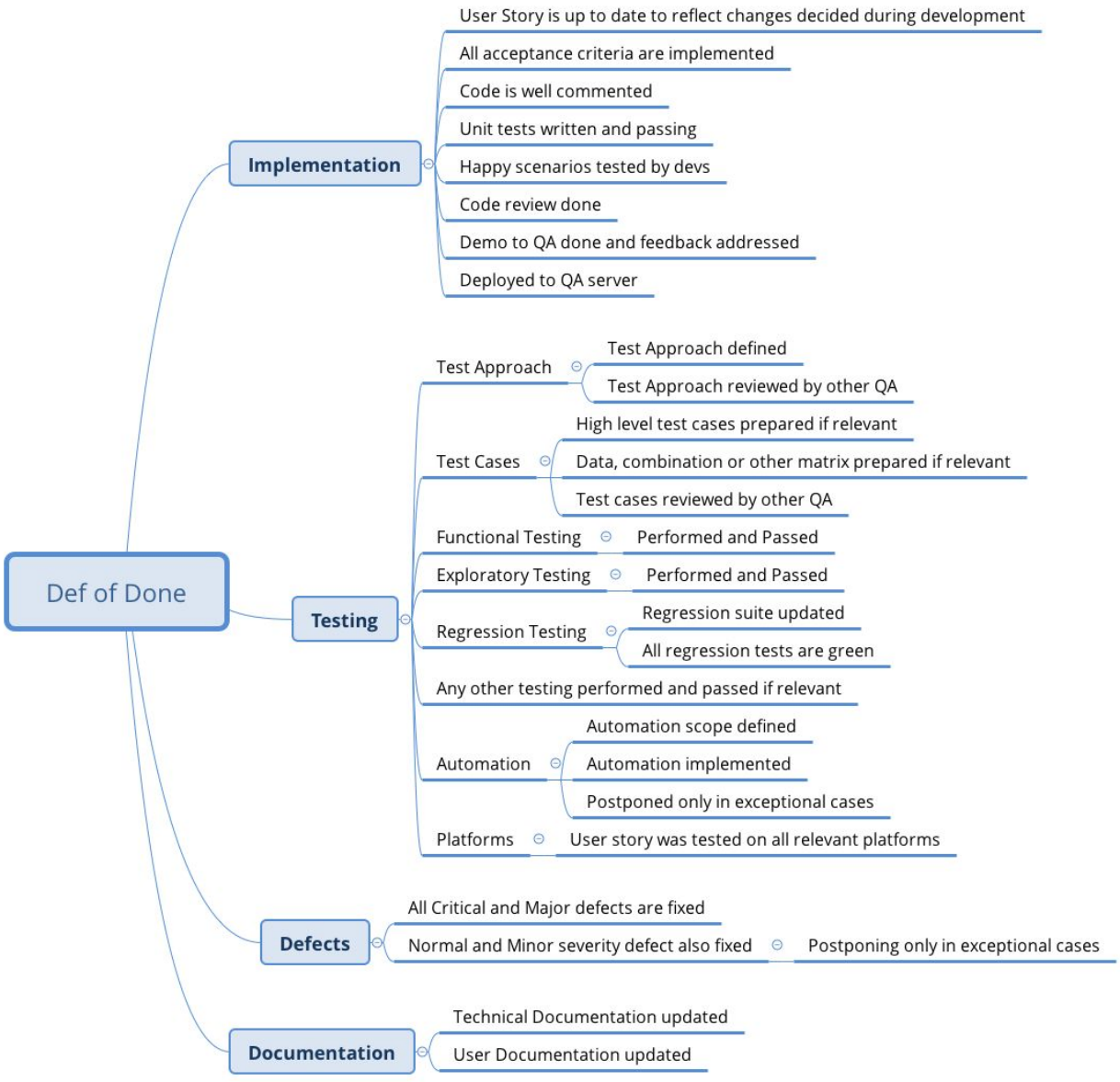
## Definition of ready

This agreement defines what has to be done before work item can be planned into the release.



## Definition of Done

Team agreement on activities and criteria to be achieved before any work item is marked as Done.



## Kanban board

Kanban board is implemented in the Polarion ALM tool when every column has agreed exit criteria to ensure the process is followed.

Release starts with planning when main targets of the Release are set and managed as Epics in Polarion ALM.

### Backlog

Epics are broken down to User Stories when every user story starts Backlog column. This column is wishlist of items to be delivered in the release but it is not consumed by the team yet.

Exit criteria

- Has top priority
- Meets Definition of Ready
- Groomed by the team = team understands the business purpose
- Sized by the team = team agrees on relative size of the item considering also potential risks, unclarity, complexity, etc.

### ToDo

This column is a queue of the items (User Stories and Defects) to be consumed by development team based on their priorities (top is highest).

Team should already have a good understanding of work items in this column but there are also additional exit criteria to be fulfilled right before Development starts.

Exit criteria

- Development strategy is clear
- QA strategy is clear
- Security impact evaluated
- Architecture impact evaluated
- Decision of test automation is made
- Discussion between Dev and QA for L and XL user stories

### Dev on-going

This column contains all items which are currently under development. Developers communicate regularly with QAs to ensure Acceptance criteria are met and also to discuss any possible changes and discoveries during the development phase. Product Owner is also invited when necessary.

Meanwhile QAs works on Test Cases to be prepared to test the feature right after the delivery. They consult test cases with Developers when relevant.

Exit Criteria

- Development finished
- Happy scenarios tested by Devs
- Main test cases prepared by QAs
- Test Cases review by different QA if relevant
- Need of exploratory and regression testing defined by QAs

## Dev Done

This column is queue of items considered to be developed but not ready to be tested yet.

Code review needs to be performed by another Developer before the item is merged to master - code reviews are managed in Git.

Demo to QA has to be performed to have a benefit of quick feedback and “right after development” discussion.

Exit Criteria

- Code review done and feedback is resolved
- Merged
- Demo to QA done and feedback is resolved

## Ready to Test

Column with all items that are ready to test. It is a simple queue without any Exit Criteria but it serves the purpose of removing bottlenecks. It has a challenging WIP to ensure every item is tested as soon as possible after development was finished.

## Test on-going

This column contains all items which are currently under test. QAs discuss their results directly with developers.

In the case defect is found new Defect Item is created in Polarion ALM, it is linked with original Item and it goes through Kanban. Original item remains in Test on-going or it goes back to Ready to Test state.

Once testing of bigger part (set of User Stories or whole Epic) is finished demo to Product Owner should be done. Additional regular demo to whole team is performed every Monday.

Exit Criteria

- Functional testing done and all test cases passed
- Exploratory testing done
- Regression testing done if relevant
- Critical and Major severity defects are fixed
- Normal and Minor defects should be also fixed. They can be postponed only as an exception and reason for it needs to be specified
- Demo to PO done if relevant

## Test Done

Column with all items that are ready to be accepted by Product Owner.

Exit Criteria

- Accepted by Product Owner
- Feedback from Product Owner implemented or planned

## Accepted

Final column from perspective of R&D. Items stays here and moves to Done Done column later in the release when final review is performed and documentation is updated.

## Done Done

All items in this column will be delivered in the release.

## Security checks done during the implementation phase

Besides code reviews which is performed on any code change there performed another activities to keep highest level of code quality.

As part of building of any component, the following steps are performed.

- Static Code Analysis
    - Java – Checkstyle and Spotbugs
    - JavaScript – ESLint
    - TypeScript – TSLint
    - Elm – Elm Analyse
- Penetration Tests
    - Newired Application Server is subjected to penetration tests using OWASP Zed Attack Proxy.
- 3rd party components audit
    - Use of major commercial and open source tools to review all included components. Fresh database of the components is always used.
- Automation done according to agreement from ToDo column

## Final Verification of the release

Every release ends with final Verification round which is performed in last weeks of the release.

Goal of this Verification is to assure that release as whole is working well, is well documented and ready to be releases.

### Feature Freeze and Code Freeze

There are two main milestones in this phase.

Feature freeze is a point of time when To-Do column has to be empty from perspective of features (user stories) - all items remaining needs to be moved back to Backlog.

Code freeze is usually week after the Feature freeze and only final regression testing and connected activities are performed after it. Every found Defect after Code Freeze is evaluated from perspective “to be fixed or not”. In the case of fixing possible impact to the release is strictly evaluated and Regression testing is repeated in the case of any risks.

### Final release audits

Final release is audited from perspective of 3rd parties risks.

- Components using Java are audited by major commercial and open source tools with a fresh database of components.
- Frontend JavaScript components are audited by JavaScript oriented major commercial and open source tools with a fresh database of components.

## Final package

Installers are packed together with documentation and installation is tested on all supported environments.

When passed this package is ready to be delivered to customers.

# Delivering releases to Customers

A new release is available to customers by the mean of link to the public folder where they can download all needed files.

New and Noteworthy document is available in the release package. This document describes what was delivered in the current release, known issues, possible security risks and other important information.

Customer Care contacts all customers to discuss upgrade and plan connected activities (help with upgrade, trainings on new features, ect.)

Customers can upgrade their environments themselves following Installation and Upgrades manual or upgrade can be performed as a service.

# Installation and Environment details

This section describes how Newired is installed on customer's premises and what are the recommended settings.

## *New installation*

Newired is installed using a Step by Step Installer

### What is installed by Installer

- Newired Application Server
    - JRE 1.8.0
        - On CentOS is used latest package from RPM.
        - On Windows is used package delivered within Newired Installer.
    - Apache Tomcat 8.5.15
        - Installer register Apache Tomcat as OS service.
        - Installer prepare Apache Tomcat working only with HTTP listening on port 8090.
    - PostgreSQL Server
        - Three databases: Newired, Report and Collector

        - On CentOS is used latest package from RPM.
        - On Windows is used package delivered within Newired Installer.
    - Newired Web application components hosted in Apache Tomcat
        - Newired Portal
        - Newired Collector
        - Newired Delivery Server
        - Newired Cross-Domain Storage Page
- Newired Editor

### Where are components installed

All Newired components are defaultly installed in folder `C:\Newired` (on CentOS it's `/opt/Newired`).

How to secure Application Server can be found in Newired Application Server Hardening document.

### How to insert Newired Snippet into Underlying Application with HTTP filter

If the customer has no possibility to insert a Newired Snippet directly into the Underlying Application, the HTTP filter can be used. The Newired Application Server Hardening document describes how to properly configure the HTTP filter for NGINX and Apache web servers.

## *Upgrade to new release - By customer*

Precondition to secure update is that the previous one was installed following steps above.

There are supported two ways how to upgrade to new release:

1. Use provided Installer.
2. Upgrade Newired Application Server manually.

Newired can help with the upgrade if necessary. It is recommended when customer setup requires update without complete reinstall.

# Customer Channels

As the first phase of Customer Support, we provide the *Customer Care* package to speed up adoption of Newired Journeys and maximize the product value from the first days after rollout. The Customer Care service includes:

- Help with installation if necessary
- Review and validate Newired installation in the customer environment (if agreed)
- Perform initial training
- Mentor the customer during their first few weeks after installation
- Review the first set of Journeys
- Collect feedback during this phase to improve the product in the future

After this initial phase, the customers are of course covered by standard Customer Support service. Customer Support is accessible via email at support@newired.com. Customers can contact Newired not only because of functional questions or issues but also in the case of any security issue or concern.

The customers are contacted by Newired in the case of any highly critical issues or high-security risks using Customer Support email channel.

The customers are informed about new releases by the mean of *What's New and Noteworthy* documentation. In addition to information about new features and improvements, it includes information about known issues and possible security risks.

All customer channels are managed in English.

## Legals

As a common practice, Newired conducts various security tests. As generally well known, security threats are continually changing, with new vulnerabilities discovered on a daily basis, and no application can ever be 100% secure no matter how much security testing is conducted. This document is intended only to provide documentation that Newired has a proper process to identify and mitigate security risks, but not to eliminate them completely. This document cannot and does not protect against personal or business loss as the result of use of the applications or systems described. Newired offers no warranties, representations or legal certifications concerning the applications or systems it tests. All software includes defects: nothing in this document is intended to represent or warrant that security testing was complete and without error, nor does this document represent or warrant that the application tested is suitable to task,

free of other defects than reported, fully compliant with any industry standards, or fully compatible with any operating system, hardware, or other application. By using this information you agree that Newired shall be held harmless in any event.