Visma.net ERP and Payroll - Technology - Security - Privacy - introduction

Visma

Visma delivers software that simplifies and digitises core business processes in the private and public sector, with presence across the entire Nordic region along with Benelux, Central and Eastern Europe.

Visma has 12000 + employees, 1.000.000+ customer contracts and a revenue of 1.5+ billion EUR in 2019.

Visma Software AS is the Business Unit for Norwegian SMB products and services.

Full disclosure

As described in this document, Visma has a comprehensive security program, VASP. (Visma Application Security Program). Most of the documentation on what we do, how and why, and status for every application regarding security, is a part of our internal management system, and not shareable to the public. Under NDA we share everything, with full disclosure, in the form of a meeting, on site, or online.

Security & Privacy

Visma has been offering cloud solutions for more than 15 years. We have established processes, methods and technologies and embraced proven standards to ensure security and accessibility for our customers. The nature of threats is constantly changing, so security awareness is a natural part of our development process and we constantly strive to be even better.

Visma has also committed to compliance with the General Data Protection Regulation (GDPR) and has for several years worked to implement policies and practices to give our customers a secure platform to trust with their personal data.

On a high level we have a Security Forum, a Security Operation Centre (SOC) and a Cyber Security Incident Response Team (CSIRT) & Coordination Center (CC) - Visma CCSIRT/CC.

The Security Forum consists of security professionals who will propose audits, alterations and additions to Visma Group Security Policies and guidelines to Visma Group management and will act as the supreme council on security related issues in Visma.

Visma Trust Centre

Visma Trust Centre gives a good overview of how we work with security and privacy. The information on Visma Trust Centre is public. Visma wants to be transparent about everything we do. However, detailed information requires a Non-Disclosure Agreement (NDA). A lot of the information that we provide requires online meetings, as the information is a part of our internal quality management systems. Visma Software AS will host online sessions if needed. We will involve relevant personnel from our Development or ITC unit, if needed for specific areas.

Product and service deliveries

All SaaS services are multitenant, and we do not provide SaaS as a private cloud solution. The products and services from Visma SMB are delivered with standard Terms of Service (TOS). The TOS is provided in our Trust Centre. It covers both on-prem and cloud software. The TOS is mandatory for SMB Software.

Visma uses only datacenters with the highest measures of security. A full overview of our data centers can be found on Visma Trust Center. https://www.visma.com/trust-centre/smb/service-information/. Visma.net ERP and Payroll is currently hosted on AWS infrastructure.

Other Business units in Visma provide Enterprise HRM solutions, with custom tailored service agreements, and Data processing agreements, with a price level aligned for enterprise delivery.

Internal security (Visma IT&C)

Visma IT&C (IT and Communications) provides all internal IT for the Visma Group, including security policies, IT services, etc.

Certifications

- ISO 9001 - Quality Management System
- ISO 20000 - IT Service Management System
- ISO 21500 - Project Management System
- ISO 27001 - Information Security Management System
- ISO 27018 - Privacy compliance in cloud

Assurance reports

- ISAE 3402 specific controls for both Visma IT&C AS and AB
  - Change Management
  - Access Management
  - Security
  - Operations
- ISAE 3000 Security controls for both Visma IT&C AS and AB
- ISAE 3402 specific controls for ASP AO services

The Visma IT&C certifications also cover operations that are provided from Visma ITC, as stated in the Scope and Boundaries.

Visma ITC is responsible for all outsourcing of PaaS (Platform as a service) and IaaS (Infrastructure as a service), both for internal usage, hosting and all Visma public cloud services. That includes all contractual matters towards our 3 datacenter partners: Digiplex (www.digiplex.com, Oslo), AWS (Amazon Web Services) and Microsoft Azure.

Software development and security

We develop, deliver and operate our cloud services based on the Visma Cloud Delivery Model (VCDM), built on industry standards and best practices. It describes aspects of how we should be organized (virtual teams, roles, responsibilities), how we should work (processes) as well as technical requirements and best practices necessary for successful cloud service delivery. VCDM is ISO 27001 certified. An ISAE 3402 for VCDM is in progress for Q4 2021.

Each Service Delivery Team (SDT) in our development units is accountable and responsible for their entire service. This includes design, development, delivery and operations, as well as security in all of these areas, e.g. each STD has his own Security Engineer.

The STD's internal security is operated from Visma ITC. Both Visma Software AS and the Development Units in Visma are ISO 9001 certified.

Independent of the VCDM, all products from Visma are onboarded to the security and privacy framework, and the Visma Maturity Index. Important highlights of the framework are:

- Security and Privacy Assessment (SSA details in appendix 1)
- Risk Assessment
- Static automatic security testing
- Dynamic Automatic security testing
- 3rd party code testing
- Manual testing (including OWASP top 10)
- Cyber Intelligence Threat program
- Bug Bounty program (with responsible disclosure, see Visma Trust Center)
- Visma Maturity Index

Some are continuous, others minimum yearly. Product individuel results and the Maturity Index can be shared in meetings.

All cloud services are secured with TLS and available by HTTPS: from supported browsers.

Appendix 1: Security Self Assessment (SSA) detailed description

(The SSA is currently in the process of a being moved to version 2.0, with a combined Security, Risk and Privacy assessment).

Visma.net ERP and Visma.net Payroll consists of several services. Each service will run through the SSA. When the SSA must be renewed version 2.0 will be applied. See Appendix 2.

Attack surface
SA01 - Attack surface interface What type of functionality does the interface provide? What kind of users (e.g. users, internal admin users, support etc.) or systems can use the interface? Insert screenshot of current Data Flow Diagram. What type of technology/protocol is used to access the interface? Examples: web service, web pages, FTP etc. How are users or systems accessing the interface authenticated? Is there any access control regulating access to the interface and the methods/resources of the interface?
SA02 - Attack surface reduction candidates Review the attack surface and note down the candidates for reducing the attack surface that you can identify
SA03 - Using up-to-date components and libraries Which components and libraries are used by the application? Examples: client side JavaScript libraries, parsers, converters, crypto libraries etc. List all components and libraries used by the application here
SA04 - Review components and libraries Are any of these out-of-date?
SA05 - Procedure to ensure that components and libraries are up to date Have routines to follow-up components so they are updated when the vendor creates a new update. Is there any procedure or solution in place to ensure that components and libraries are kept up-to-date? See OWASP A9 Using Components with Known Vulnerabilities for more information
General input validation practices
SA06 - Client side input validation Does the application rely on client side input validation?
SA07 - Input validation coverage and quality Is input received via all interfaces of the attack surface validated before it is processed/persisted?

Is input validation centralized in one component or is it implemented independently in all interfaces?

Identity all code fragments or components that implement input validation. Review the input validation code of at least a sample

SA08 - Input validation coverage

Are all types of input validated, including for example cookies and HTTP headers?

SA09 - Validation extensions and uploaded files

Does the application allow that files (e.g. report definitions, attachments) or extensions/code are uploaded to the site which is later executed or parsed server side?

If so, how is it validated that the uploaded content is safe?

If files are stored before they are validated are they stored i a safe location (i.e. not in a way so that it can overwrite web pages or be accessible as a web page)?

More on this subject

SA10 - Insecure redirects

Does the application redirect to other sites or internal web pages. If so, is the redirect destination based on user input and if so how is this input validated?

Does the application have a filter that check that redirect is only made to white listed sites?

Example:

The LogIn action in the Account Controller does not reject absolute URLs in the ReturnUrl parameter.

http://www.app.com/Account/LogIn?ReturnUrl=http%3A%2F%2Fwww.evilsite.com

This vulnerability could be used in a phishing attack.

OWASP Unvalidated Redirects and Forwards Cheat Sheet

Session management

SA11 - Custom session management

Is session management handled by the framework or is it custom built?

If custom built review the code. Are session ids long and random or can they be guessed/predicted? Is the format of the session id validated by the server? Is the session id killed when the user logs out? etc. Review against OWASP Session Management Cheat Sheet.

SA12 - Session expiration

When and how is the user session expired?

Is all cache cleared as part of the expiration?

For more information see OWASP Session Management Cheat Sheet

Access control

SA13 - Client side access control

Does the application in anyway rely on client side access control?

Is information about current user and authorizations retrieved from a trusted source? I.e. not from the client.

SA14 - Access control coverage and quality

Are all access requests (e.g. HTTP request, web service calls) subject to access control?

Review the access control code of at least a sample of the interfaces of the attack surface

SA15 - Fail securely

How is access control applied to all resources (pages, services etc.)?

Is it injected through manual coding or automated through injection, aspects, interceptors or similar mechanism?

What happens if a developer forgets to configure access control rules for a resource?

Does it fail securely? I.e. access is denied for all.

For more information see OWASP Fail Securely

SA16 - Insecure direct object references

Does the application use direct object references and/or indirect object references?

If direct object references are used are authorization checked when the request is returned?

An explanation of Insecure direct object references

SA17 - Unvalidated forwards

Does the application forward the user to internal pages based on user input?

If so, how is this input validated and is access control invoked when the page the user is forward to is requested?

When applications allow user input to forward requests between different parts of the site, the application must check that the user is authorized to access the URL, perform the functions it provides, and it is an appropriate url request. If the application fails to perform these checks, an attacker crafted URL may pass the application’s access control check and then forward the attacker to an administrative function that is not normally permitted.

An explanation of Unvalidated forwards

OWASP's Unvalidated Redirects and Forwards Cheat Sheet

Multi-step business flow

SA18 - Multi-step business flow integrity

Does the application support processes that consist of several steps that must be performed in a certain order and may there be something to gain by skipping steps or changing the order of the steps?

If so, how does the application ensure that the steps are performed in the specified order?

For more information see OWASP Testing business logic

SA19 - Multi-tenancy

How are tenants/customers isolated from each other?

How are sessions of users connected to the tenant/customer id or database?

Database level: each tenant/customer has a separate database and the application ensures that the correct database is accessed for each request.

Application level: tenants/customers share database and application ensures that the correct tenant/customer id is included in each database request.

Review code to ensure that it is not possible for a user to gain access to the data of a tenant/customer that he is not authorized to access.

Injection Prevention

SA20 - Dynamic SQL

Are there any SQL queries that are created by concatenating strings to SQL statements?
If any are found, is this necessary?
Search the code for execution of concatenated SQL statements.
If it is necessary, review these SQL statements in detail to ensure that they are safe.

SA21 - Other forms of injection

If the application uses reflection to load classes and execute methods based on user input, review the code to ensure that the call flow can not be manipulated (Reflection injection)?
If other types of subsystems are used and called based on user input, review the code to ensure that the input is validated and sanitized/encoded before it is passed to the subsystem.
Example : Xpath injection

Cross Site Scripting (XSS) Prevention

SA22 - Server side: Output encoding of data included in dynamic webpages

How is the framework(s) used to build web pages used to ensure that cross site scripting vulnerabilities are not introduced?
Confirm that the used framework(s) does encoding?
At least review samples of code that builds webpages and search for code that circumvents the framework by injecting raw data into HTML, script tags etc.

SA23 - Client side: Encoding of data included in dynamic webpages

Is the DOM manipulated by JavaScript?
How is is encoding ensured; By the framework or by the application using standard encoding libraries?
At least review samples of JavaScript code that manipulates the DOM.

SA24 - Content security policy

Is Content Security Policy used to restrict use of JavaScript (e.g. inline JavaScript that might have been injected through a XSS vulnerability)?
See Content Security Policy presentation

Cross site request forgery (CSRF) prevention

SA25 - Cross site request forgery protection

Are form tokens (e.g. AntiForgeryToken), the HTTP-Referrer header, re-authentication and/or any other mechanism in use to protect against CSRF?
If the application does not protect against CSRF which actions can an attacker potentially get an user to perform? How security critical are these actions?
Sample: Re-authentication should be required when users request to change password.
If the application protects against CSRF review at least that it is implemented correctly for the most severe actions.

SA26 - Protection against Clickjacking

Does the application protect against Clickjacking?

State management/Client-side storage

SA27 - Client side state storage

Is state stored in cookies or other client side mechanism (e.g. viewstate, hidden field)?

If so, does the application depend on the integrity of this data?

If so, how is it protected against tampering?

SA28 - Secure cookies

If applicable, is the security cookie attribute set to true on each type of cookie so that cookies is only transported over HTTPS?

SA29 - HttpOnly attribute

If applicable, is the HttpOnly cookie attribute set to true on cookies so that JavaScript can not access the cookie (especially important on cookies that store session ids)?

For more information see OWASP HttpOnly

SA30 - Cookie expiration

Is persistent cookies is used?

Are users informed that persistent cookies are used and for what?

If so, what data is stored?

Is there any potential confidentiality issues?

Is the data in the persistent cookie validated and encoded before being used?

Is there any expiration date set on the cookies?

For more information see Cheat Sheet

SA31 - Browser cache

Is the browser cache used?

Is use of the browser cache turned off by default?

Is the user informed of the consequences of turning on the browser cache (e.g. the user might access you application from a public computer)?

Is the user informed of what is stored locally?

If so, what data is stored?

Is there any potential confidentiality issues?

Is the cached data validated and encoded before being used?

SA32 - HTML 5 Client Side Storage

Is the application using HTML5 Client Side Storage (localStorage)?

Is use of HTML5 Client Side Storage turned off by default?

Is the user informed of the consequences of turning on client side storage (e.g. the user might access you application from a public computer)?

Is the user informed of what is stored locally?

What kind of information is stored?

Is there any potential confidentiality issues?

Is the stored data validated and encoded before being used?

Have you implemented any expiration policy?

SA33 - Browser specific Client Side Storage

Is the application using browser specific Client Side Storage like SQL-Lite built in support in Chrome?

Is use of browser specific Client Side Storage turned off by default?

Is the user informed of the consequences of turning on the browser cache (e.g. the user might access you application from a public computer)?

Is the user informed of what is stored locally?

What kind of information is stored?

Is there any potential confidentiality issues?
Is the stored data validated and encoded before being used?
Have you implemented any expiration policy?

Privacy and data protection

SA34 - Data list
A list of personal data that is processed by the application/ service at the database/ data field level.
For example: name, address, date of birth, account number, IP address.
This should be available from the system documentation- feel free to just link to that.
Please note that this list is something customers regularly ask for in Trust Centre Level 3 requests, and it may be very difficult to do the next step without this list.
NOTE: obviously, if the customer, partner etc define the fields themselves, you cannot answer this.

SA35 - Risk Assessment (incl Data Classification)
Please complete the Risk Assessment →

SA36 - Personnel
All employees are expected to go through the basic e-learning course. "Specialist training" can be tailored to each role depending on need and context, and there's a specialist course in Privacy by Design and Engineering (not ready at this time).If you have questions about training, please contact Lars Holtar, data protection manager

SA37 - Third parties
By "third parties" we mean non-Visma entities that are somehow involved in processing the data in question.
Examples of third parties are Amazon for AWS, Hotjar and Google Analytics or Sendgrid for emails, or consultants.

SA38 - Access and Authorisation
No anonymous user accounts or shared/ system users should have access to personal data.

SA39 - Data Retention and Deletion
It is very important to know what data must be deleted, and when, and what data must be retained.
In general, customer-owned production data should be deleted after a certain period when the customer relationship ends. This period is stated in the contract (e.g. the Visma.net TOS).
Visma-owned data, such as customer records, invoicing/ accounting data, shall be retained for a specified period determined by the purpose for processing and legal requirements.

SA40 - Data Return and Data Portability
When Visma hosts the customer's data, the customer may request a copy of it, typically when the customer relationship ends. This is part of the contract with the customer.
The deletion of customer data happens after that return of data, and a grace period during which the customer may check the data's integrity.

SA41 - Data Rectification

Personal data should always be up to date, relevant to the purpose (identified in SA35) and accurate. This is the responsibility of the data controller, which typically is a customer. It can also be an individual data subject.

Our products and services should enable the data controller to keep personal data up to date, relevant and accurate.

SA42 - Reservation and withdrawal of consent

An individual has the right to reserve against certain types of processing. The most relevant is targeted marketing and profiling. Similarly, the individual has the right to withdraw consent to such processing.

SA43 - Notification and permits

This is not about access to the data by the authorities, but wether or not the data controller must notify the authorities or data subjects about the processing, or seek permission from authorities for the processing.

In most cases, notification and permits will not be required.

Examples:

Notification to customer/ data subjects if sensitive data is moved out of the country (e.g. if we move Visma.net Expense hosting services to another country) Certain public sector processing relating to sensitive personal data.

This information is important for building the Trust Centre and enabling our customers to comply with the law.

SA44 - Cookies and similar

Online identifiers such as IP address that is collected through a user's browser using cookies or other technologies, is defined as personal data. It is important that this data is only used for authorised purposes.

Authorised purposes can be found in the Visma Privacy Statement and/ or customer contract (e.g. Visma.net TOS).

SA45 - Encryption and signing

Personal data should as a general rule be encrypted, especially for higher risk levels.

SA46 - Crypto algorithms and key management

Which encryption algorithms are used by the application?

How long keys are used?

How are the keys managed (i.e. established, stored, recovered, changed etc.)?

SA47 - Privacy by Design

"Privacy by Design" are a set of design- principles that are part of the GDPR.

Compliance with these principles are partially acheived through certain other SSA-items (please refer to 1.2. SOA: General, Privacy by Design), however, "yes" here is required for Level II compliance.

Non-repudiation controls

SA48 - Non-repudiation controls

Is the application used for making transactions (e.g. e-commerce, banking) and it is important that transactions can not be denied?

If so, how does the application protect against repudiation of these transactions?

Security monitoring capabilities

SA49 - Security logging

Are security events logged to a security log?
What is logged? Time, user and/or IP address and event should at least be logged.
Examples: logins, changes to role assignments, password, bank account numbers etc.

SA50 - User access to security events

Are information about relevant events made available to customers and users (e.g. changes to bank account numbers)?

SA51 - Integrity protection of logs

How is the integrity of the security log protected?
Which type of threat agents are protect against; insiders, outsiders/hackers and/or users?

SA52 - Application misuse detection

Does the application have any intrusion/misuse detection capabilities?

Error handling

SA54 - Handling errors

Does the application use custom standardized error pages that do not expose application details for both handled and unhandled errors?
Review the application to ensure that these types of non-detailed and user friendly error messages are returned for all errors.

Secure User Notification

SA55 - Phishing

Does the application generates messages or notifications that are distributed via email or other form of messaging that contain links to log-in pages?
For more information on phising see
http://www.microsoft.com/security/online-privacy/phishing-symptoms.aspx

AJAX

SA56 - AJAX security

Does the application use AJAX or other dynamic web technologies?
If so, review the application against OWASP AJAX Security Guidelines.

HTML5

SA57 - HTML5 security

Does the application use HTML5?
If so, review the application against OWASP HTML5 Security Cheat Sheet.

Secure Cross-Domain Requests

SA58 - Cross-domain requests

Is the application is using or accepting Cross-Origin requests using JSONP?
JSONP has several potential security problems.
For example:
CSRF vulnerabilities. You have to remember to defend against CSRF vulnerabilities, and with JSONP, that gets a bit tricky. Standard advice is to ensure that only POST requests can trigger a side-effect, and to include a CSRF token in all POST requests; but JSONP involves sending a GET request to trigger a side-effect, which ain't exactly the cleanest

solution you've ever seen. So this means that the host that provides JSONP service needs to remember to check CSRF tokens even on GET requests. Also, it requires a bit of a tricky protocol for the embedding page (a.com) to obtain the proper CSRF token from the JSONP service (b.org). It gets messy.

Web Service and REST Security

SA59 - Web service security
Does the application exposes web services?
If so, review them against OWASP Web Service Security Cheat Sheet and OWASP REST Security Cheat Sheet.
(Please note that concerns such as access control, input validation, injection prevention etc. should already have been reviewed for these web services as they are a part of the attack surface.)

Account and password management

SA60 - Provisioning of accounts and passwords
Review against the "Provisioning of passwords" section of Password management guidelines

SA61 - Handling of passwords
Review against the "Handling of passwords" section of Password management guidelines

SA62 - Controls to prevent or limit consequence of compromise of passwords
Review against the "Controls to prevent or limit consequence of compromise of passwords" section of Password management guidelines

SA63 - Controls to detect compromised accounts
Review against the "Controls to detect compromised accounts" section of Password management guidelines

Review of Deployment

SA64 - Firewall configuration
Are the firewall rules for the application documented and verified?
Inbound: On which ports should incoming calls be accepted to the application and its subsystems and from where should calls to each port be accepted (e.g. Internet, integrated systems)?
Outbound: Which other services must the application and its subsystem be able to connect to?

SA65 - Dynamic page engines
Does the web site require dynamic page engines like .NET, PHP, CGI?
If so, verify that only required engines are enabled.

SA66 - Application permissions
Is it documented which permissions the user the application is executing under needs (Folder and file access and OS privileges)?

SA67 - Database permission
Is it documented which permission the application need in the database?
Upgrade of the application often require more database permissions than the application need. Is a separate database user used for database upgrades?

SA68 - Other subsystems
If the application uses other subsystems, is it documented which permissions the application need in these subsystems?
Is it documented what permissions do the users the subsystems are executing under need (Folder and file access and OS privileges)?

SA69 - Integrations
If the application is integrated with other systems, is it documented which permissions the application need in these systems?
If other systems are integrated with the system, which permissions does these system need in the application?

SA70 - HTTPS/SSL
Is it documented which incoming connections should require HTTPS/SSL?
HTTPS/SSL must be used for all communication over the Internet.
For all incoming connections that require HTTPS/SSL, regular HTTP should be disabled.

SA71 - Keys
Does the application depend on keys to protect the integrity and/or confidentiality of data?
If so, is it documented which keys the application must be configured with?
Is it documented how the confidentiality of these keys should be protected?
Is it documented how these keys can be changed?

SA72 - Credentials
Does the application depend on stored credentials to authenticate to subsystems and integrated systems?
If so, is it documented which credentials the application must be configured with?
Is it documented how the confidentiality of these credentials should be protected?
Is it documented how these credentials can be changed?

SA73 - Robots.txt
Is it defined which part of the application that should be indexed by search engines?
For more information see http://en.wikipedia.org/wiki/Robots_exclusion_standard

Apppendix 2: Security self assessment 2.0

As mentioned in appendix 1.0, a new SSA template is ready, and products are onboarded when SSA is outdated, and it is used for all new products. Earlier the security, privacy and risk assessments were divided into different templates. In SSA 2.0 the three topics are combined. The structure of the new templates is described below:

RM01: Risk Profile
SEC01: System Diagram
DP01: Data List
DP02: Data Classification
DP03: Privacy and Data Protection by Design
DP04: Formal requirements and standards
DP05: Customer contract and supported version
SEC02: Attack Surfaces
SEC03: Access Control Quality
SEC04: Password storage
SEC05: Crypto/hash algorithms
SEC06: Application misuse
SEC07: Software Dependencies
SEC08: File upload validation
SEC09: Secrets in source code
SEC10: Secret Management
SEC11: Phishing
SEC12: Testing and Quality Assurance
SEC13: Secure Deployment
SEC14: Infrastructure permissions (databases, storage, queues, service bus etc)
SEC15: Host and Network Security basics
SEC16: Security Logging
SEC17: Threat Intelligence
RM02: Risk Review
RM03: Risk Assessment
RM04: Risk Register

Appendix 3: Visma.net ERP Architecture

The application is an ASP.Net based web application.

Front-end

The front-end is based on ASP.Net web forms. All pages is based on ASP.Net web forms. The navigation part of the application where you have the menu, workspaces, tiles and search is based on React framework.

We have decided to use React library as front-end framework for future architecture and new screens we will create.

The front-end sends and receives information from back-end where the communication is based on callbacks and events.

All UI controls are own developed controls and mapped to server controls like PXFormView, PXGrid, PXSelector and PXText.

Browser will cache page content (html) in browser cache and it will not get html and java scripts on every request.

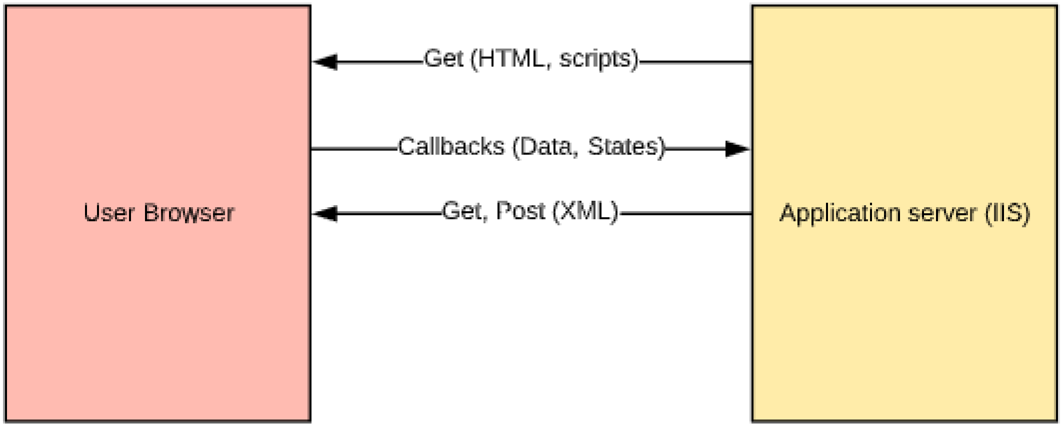
Visma.net ERP is encrypted in transit with https and TLS 1.2 or newer.

Back-end

The back-end is based on .Net Framework.

On the server side all pages are connected to a business logic container - graph. All container controls are linked to caches (DAC) - data view by data member property in the pages (aspx).

All business logic is event based and triggered by events.



Overview of services and technology used

Module/Service	**Platform**	**Comments**
Application(web) server	IIS - ASP.Net,, React Net Framework 4.7.2, C#	Multi-tenant. REST API, Redis cache service
Visma.net Dispatcher	.Net Framework 4.7.2, C#	Message dispatcher - communicates with Message Router (ActiveMQ)
Visma.net Downloader	.Net Framework 4.7.2, C#	Downloads new invoices from AutoInvoice
System Data Service (SDS)	.Net Framework 4.7.2, C#	Our centralized service who keeps information about companies, users, databases, configuration data and license information
Visma.net.ERP Connect (Selector)	.Net Core 2.0	Authentication, integration with Visma Connect
Visma.net.ERP Provisioning Engine (VEPE)	.Net Core 2.1	Deployment, Upgrade of new versions, provision of new companies, new users and role assignments. Integration with Provision Gateway. REST API
Approval Editor	.Net Core 2.1 , front-end: Angular 1.x	Editor for supplier documents for users within Approval
Visma.net.ERP Common Data Service	.Net Core 2.0	Store data which is common for all companies; Currency data - exchange rates - fetched from AutoPay
Scheduler Dispatcher	.Net Framework 4.7.2, C#	Trigger scheduling tasks - automation of processes
Sigma Sync Engine (MDM integration)	.Net Framework, C#,MSMQ	Integration with MDM

Visma.net ERP CopyTemplate	.Net Framework, C#	Used to copy information from one company to another one - usually template companies
Database	MS SQL	Today: One database for each customer. In progress: Consolidating all databases into multi tenant databases so will have full multi-tenant on database too. We will have different databases for each country and seperate database for DEMO and Test customers. Sharding of databases will be based on tenantid- which tenant has which database is stored in System data service

Appendix 4: Backup and restore

Visma has a full set of procedures for handling incident management and a business continuity plan in case of emergencies.

Visma.net ERP follows the following Backup regime.

Log backup x 2 times pr. day

Full backup x2 times per week.

Retention is 30 days (point in time restore possible)

Monthly backup once a month

Retention is 12 months

Yearly backup once a year

Retention is 10 years

Appendix 5: User identity

Visma has its own user identity solution; Visma Connect, as a Single Sign On (SSO) service for all Visma services, and for our common landing page Visma Home.

Here is a list of Visma Connect features that is commonly required:

- Visma Connect supports MFA (sms, BankID, hardware key, Authenticator apps)
- FIDO2 support / Passwordless login
- Password complexity for common users i minimum 8 characters, large, small, numeric and special. 14 characters for admins.
  - A new feature for customized setup per customer common added in Q3 2020.
- IP, browser and more customizable limitations added in Q3 2020.
- AD/SAML integration, currently in pilot with selected customers. Public available planned but with assisted setup
- OpenConnect / 0Aut supported
- All users are managed by customers administrators in the role based function Visma.net admin.