MPP.360 Platform

Statement of Compliance

effective

October 01, 2023

Contents

1 General Information

1.1 Scope

This document refers to the software “MPP.360” and explains the technical outline of the software solution. MPP.360 is designed and implemented with a service-oriented architecture in mind. Therefore, individual parts of the solution (“modules” / “services”) are seen and treated as individual software services. Each module is isolated in its own physical runtime environment and persists its data separately onto its own physically isolated database and file storage system. The main module, “MPP.360 Core” communicates with these services by secured REST-API-Calls.

1.2 Definitions

Contractor: “DIU MarTech Solutions GmbH” as the sole software development and integration partner of MPP.360.

Infrastructure: Cloud environment(s) and services provided by AWS.

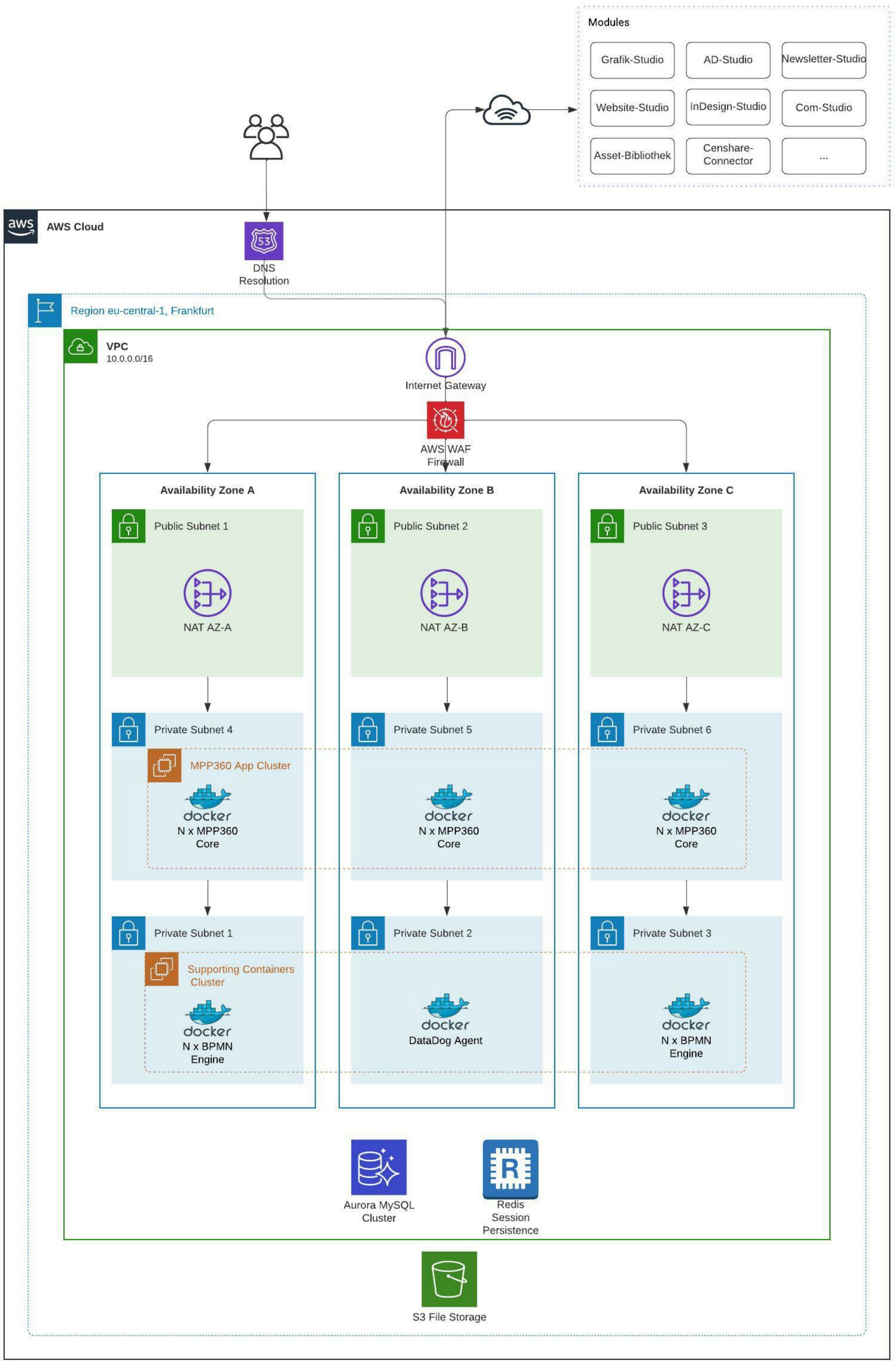
2 Solution Architecture

2.1 AWS

MPP.360 is built with AWS technologies as a cloud-native software solution. To maintain the product’s conformance with the European General Data Protection Regulation (GDPR, DSGVO, Datenschutz-Grundverordnung), only AWS services offered in the European zone, mainly Frankfurt/Germany are used.

2.2 MPP.360 Core

The “MPP.360 Core” is a platform that offers the core functionalities needed by the organization, like user management, job handling, workflow design, informing users via notifications and announcements, role-based access to briefing and production data. Along with these responsibilities, it orchestrates available modules to offer a unified and integrated user-experience to the end-users. Since the core module is capable of receiving inbound connections via its API and is also enabled to call external APIs on various events via webhooks, third-party systems and services can be easily integrated with MPP.360. This level of interoperability has been reached by employing a microservice architecture in which every single module is highly available and individually scalable. The diagram below shows the high-level architectural design of MPP.360.


Modules
Grafik-Studio
AD-Studio
Newsletter-Studio
Website-Studio
InDesign-Studio
Com-Studio
Asset-Bibliothek
Censhare-Connector
...
AWS Cloud
DNS Resolution
Region eu-central-1, Frankfurt
VPC
10.0.0.0/16
Internet Gateway
AWS WAF Firewall
Availability Zone A
Availability Zone B
Availability Zone C
Public Subnet 1
Public Subnet 2
Public Subnet 3
NAT AZ-A
NAT AZ-B
NAT AZ-C
Private Subnet 4
Private Subnet 5
Private Subnet 6
MPP360 App Cluster
docker
N x MPP360 Core
docker
N x MPP360 Core
docker
N x MPP360 Core
Private Subnet 1
Private Subnet 2
Private Subnet 3
Supporting Containers Cluster
docker
N x BPMN Engine
docker
DataDog Agent
docker
N x BPMN Engine
Aurora MySQL Cluster
Redis Session Persistence
S3 File Storage

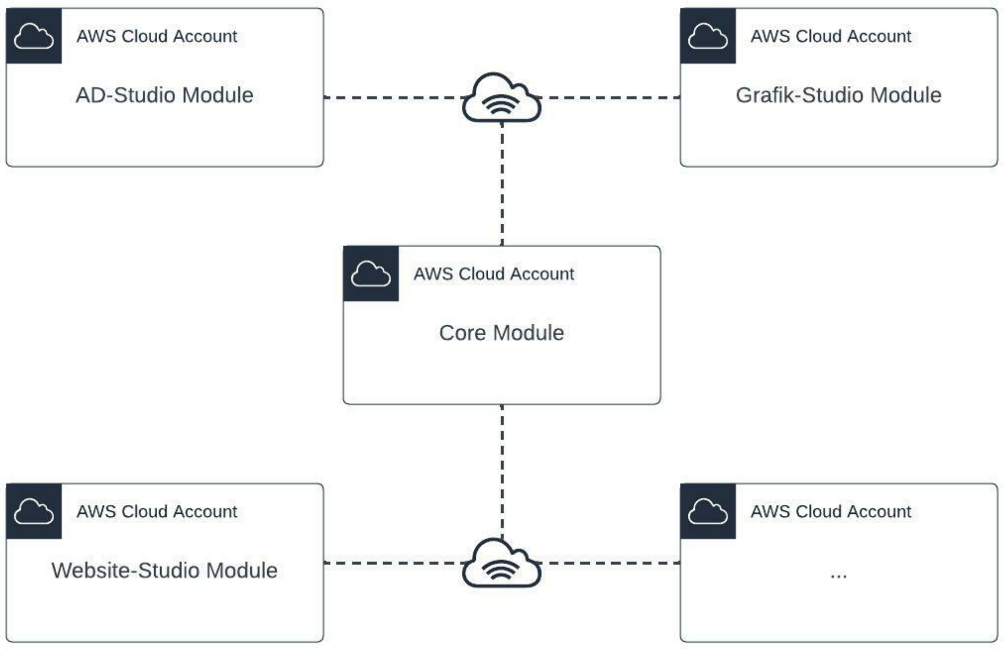
2.3 MPP.360 Modules

The modules listed below are currently available:

Grafik-Studio	editor to create and edit graphical assets
AD-Studio	editor to create animated HTML banners
Newsletter-Studio	editor for creating newsletters in the given template
Landing-Page-Studio	editor for generating landing pages, websites
InDesign-Studio	integration for Adobe ™ InDesign to generate PDF documents
Pattern Studio	Defines processes, dynamic forms, call to external APIs via webhooks
Censhare-Studio	integration to connect with the Censhare ™ platform
Com-Studio	gateway to deliver transactional email with individual templates
Export-Studio	microservice to export marketing collaterals to non-API services like S3

2.4 Dedicated Infrastructure per Module

While modularity and separation of concerns at the software level have been designed and implemented, MPP.360 also uses dedicated AWS accounts for each individual module. The diagram above depicts this isolation.



By guaranteeing such a level of isolation for each module at the infrastructure level, updating a module or its underlying infrastructure doesn't have any impact on the rest of the system.

So, maximum security for the tenant's data with respect to accidental data losses has been achieved.

3 Cloud Architecture

3.1 Architectural Principles

The design and development of MPP.360 strictly emphasizes leveraging AWS cloud native services and architecture patterns for maximum scalability, security and standard compatibility. A plenty of business functions are deployed on AWS Lamba following serverless architecture principles. In case a service should be in constant use of end-user, those applications are containerized and deployed on AWS ECS clusters. The system design follows best practices as suggested by AWS Well-Architected Framework. Architectural design quality is governed using AWS Well Architected Tool.

3.2 AWS services

Compute	
service	comment / usage
Amazon EC2	worker nodes for containerized services
AWS Lambda	microservices and serverless applications

Containers	
service	comment / usage
Amazon ECR	private docker image registry
Amazon ECS	cluster & container orchestration

Storage	
service	comment / usage
Amazon S3	binary file storage
AWS Backup	automized backup of AWS services data

Database	
service	comment / usage
Amazon DynamoDB	schemeless DB
Amazon Aurora DB Cluster	RDBMS, compatible with MySQL
Amazon ElastiCache for Redis	session persistence

AWS DocumentDB	Graphic Studio Component

Security, Identity & Compliance	
service	comment / usage
AWS Identity & Access Management (IAM)	used for SSO logins to AWS accounts / API programmatic user accounts for deployments over BitBucket pipelines
Amazon Cognito	federated Identities, multi-factor authentication
AWS Secrets Manager	data store for API keys
AWS WAF	web application firewall
AWS Security Hub	security scans of the whole infrastructure with “AWS Foundational Security Best Practices” and “CIS AWS Foundations Benchmark” measures.
AWS GuardDuty	Intelligent thread detection
AWS Inspector	Automated vulnerability detection
Bucket AV Virus Scanner	Service to scan on S3 user uploaded files for known viruses/malware based on ClamAV

Cryptography & PKI	
service	comment / usage
AWS Certificate Manager	management of SSL certificates incl. automatic renewal.

Machine Learning	
service	comment / usage
AWS Translate	Multi-Lingual support in UI

Management & Governance	
service	comment / usage
AWS CloudTrail	audit logging of the infrastructure components

Amazon CloudWatch	used to schedule periodic system events and logging business events
AWS Control Tower	used to rollout infrastructure wide policies
AWS Management Console	GUI for managing cloud resources
Systems Manager	SSH keyless connection to worker nodes (SSM) - store sensitive data in parameter store

Networking & Content Delivery	
service	comment / usage
Amazon API Gateway	public API endpoints for microservices and modules
Amazon CloudFront	public CDN
Elastic Load Balancing	to distribute load between worker nodes in ECS clusters
Amazon Route 53	DNS to manage hosted zones
Amazon VPC	used to build network infrastructure

Application Integration	
service	comment / usage
Amazon SNS	send notifications via different channels, like E-Mail
Amazon SQS	used to enqueue workloads for async. processing
AWS Step Functions	implement state machine logic using serverless

Business Applications	
service	comment / usage
Amazon SES	E-Mail reception and delivery

3.3 Development Tools & Services

usage	service
infrastructure as code	Terraform
sourcecode repository & versioning	Bitbucket
operations, issue- and product management; documentation	Jira, Confluence,
user documentation	Bookstack
API documentation	Swagger/UI
application. & service monitoring	DataDog
Static Code Analysis	Snyk

3.4 Programming Languages & Frameworks

Different modules of MPP.360 are built and developed using different and actively maintained programming languages. The selection of a programming language follows internal guidelines and depends on the following criteria:

- Ability to get integrated with modern cloud-based resources
- Active and large support community
- Continuous security updates
- Advanced level of know-how by minimum 2 developers in team

4 Development and Delivery

MPP.360 has employed agile methodologies such as Kanban to develop and deliver software solutions, while promoting extensive communication between individuals in the project, close collaboration with the clients and responding to requested changes in an efficient manner.

4.1 Development Model

To provide a clear state for each module of the product, the Kanban methodology is used. Each unit of work has an explicit state of “todo”, “doing” or “done” where the key measure of progress is the working software as expected. The development team is accompanied by a Product Owner where the PO acts as a proxy between the development team and MPP.360 organization.

4.2 Continuous Delivery

While using Kanban provides a clear and concise workflow for each unit of work, Feature Driven Development (FDD) has also been adopted to provide continuous delivery in scheduled iterations. In every iteration, a weekly release is scheduled, developed, tested, and delivered. Each delivery package contains new features, bugfixes and improvements and the iteration are documented and communicated via release notes to all stakeholders.

4.3 Source Code Management

Git is used, hosted on BitBucket for the source code management. For the branching model, GitFlow has been adopted. Every developer holds the least-privileges that she needs for her development on the source code.

4.4 Operational Environments

To develop, test and release the software, the following environments are in use:

- Local: Isolated offline environment for each developer, to code and debug
- Testing: Isolated online environment in which the developers deploy their latest code for PO review
- Staging: Isolated online environment on which, every concrete release is deployed and tested before releasing it to production use
- Production: Isolated online environment in which the latest version is available for production use

In the software development process, no end-customer data of any kind is used.

4.5 Deployment

4.5.1 Software

Deployment procedure is automated by pushing on specific Git branches leveraging AWS services for container building and roll out. To prevent accidental deployments, production branches have been set up as protected branches. Also, every single feature which has to be delivered for the PO review must pass a peer-review in which minimum one other developer checks the code changes of the code-author.

4.5.2 Infrastructure

Cloud infrastructure follows infrastructure as code concept and is implemented using Terraform. Separated code repositories are in use which not only allow automation of infrastructure, but also prevent accidental infrastructure changes.

4.6 Test and Quality Assurance

To assure the soundness and stability of each version, the following quality assurance measures are in place:

- End-to-end tests by the PO for each single feature
- End-to-end tests of the whole release
- After each release, the new features and the essential functionalities are tested in the production environment.
- Pair-programming by conception/implementation of complex features
- Peer-review done by the development team before delivery on the testing environment.

4.7 Documentation

Topic	Target group	Format	Goals & Contents

Infrastructure	Developers, Auditors	Digital graphical: UML, Context	technical reviews architecture, employed components, networking model, security layers and information flow at the hardware level.
API	Developers, Integrators	SwaggerUI, Open API 3	Enable integrators to leverage MPP.360 and integrate into business processes.
Integrator Documentation	Integrators	Bookstack	release notes, Q&A, feature documentation
Source Code	Developers	clean-code guidelines main logic and flow of the use-case is also being documented at the class and method levels	a high degree of readability

4.8 Software Development Principles
By the design, development, maintenance and update of all software modules, clear principles like modular design, object orientation, separation of concerns etc. have been defined and implemented. Also, the control on the compliance of implementation with these principles takes place constantly, as a part of code-reviews and pull-requests. In case of non-conformity detection, the module in question would be send back for correction.

4.9 Communication and Coordination
To communicate the requirements from the end-user level to the development-end, a number of meetings and coordinations between the end-customer and MPP.360, intenally in MPP.360, between MPP.360 and the contractor and internally in contractor's teams is schedueld and held regularly. By the communication, a special attention to security, need-to-know principle and also legal aspects of requirements/technical approaches is paid.

5 Security
MPP.360 is built with security-first principle in mind. This applies to all aspects of the solution, every single component, code, hardware, and staff. The following describes how security is perceived and implemented, which governance models and benchmarks are used.

5.1 Software Security
At the software level, following best practices and standards is constantly encouraged and on the other hand, inventing self-made security solutions is highly discouraged. This applies to different aspects of the software including handling user input, cryptography, detecting

anomalies, etc. The following subsections explain security principles implemented by MPP.360 software.

5.1.1 Authentication and Authorization

MPP.360 interacts with users at two different levels. The front end and the API level. Each action with the frontend requires an authenticated user, having the adequate right for the requested action.

At the API level, each request needs an authorized access key and in case the request involves an update operation, the access key must explicitly allow a write permission.

At both, the frontend and API level, users are directed to an isolated tenant environment after a successful authentication or authorization.

5.1.2 Brute-Force Detection and Prevention

MPP.360 can detect Brute-Force attacks, I.e., when the attacker tries to guess credentials of a user by performing automated login requests. To prevent such an attack, the system enters the human check mode and expects a strict captcha verification on each login request.

5.1.3 Role Based Access Management

MPP.360 is a role-based system in which unlimited roles can be defined dynamically. Each role can then become the privilege to access a specific part of the system. So, authorized access by each single transaction from the user to the system is guaranteed.

Each user in MPP.360 will have a minimum of one role. Also, one can acquire additional roles by a user-admin. An Admin in the tenant's account can define and regulate these roles, and their assignment to the users.

5.1.4 IP-Based Protection

Upon the request of the client, it is possible to define parts or the whole environment of the client as IP-based protected. This is useful when the client wishes to use the software only from trusted IP addresses, like the organization's VPN.

5.1.5 Federated-Identities

Besides the native authorization scheme, MPP.360 supports a wide range of identity providers, public as well as private providers. A tenant's environment can be configured to use the organization's own authentication point in order to make users login into the system. This has been done by supporting standard authentication methods like SAML2. Doing so, users in an organization benefit by using only their own organizational credentials to work with MPP.360.

5.1.6 Multi-Factor-Authentication

Having federated-identities supported, it is also possible to force the users to perform Multi-Factor-Authentication upon logins. This can be done at the organization's authentication point, if the tenant's environment is configured to use the organization's authentication scheme.

5.1.7 Unique and Single-person User Accounts

To identify the end-users who interact with the system in a secure manner, having “group user accounts” or “shared user accounts” are not allowed and not supported.

5.1.8 Support of User Account Deactivation / Deletion

Upon a user’s leave from the system, there is the possibility to deactivate or delete the user’s account in question.

5.1.9 Secure Communication regarding User Accounts

In the event of creating a new user account or resetting a user account’s password, a system-generated link is sent to the user in question securely via E-Mail. The user is then able to choose a password via the provided link. The only entity that knows about the chosen password is the user herself and no admins can see the selected passwords.

5.1.10 Controlling Admins and Management User Accounts

There is the possibility of having more privileged user accounts in the system. For example, integrators, advanced integrators or admins. Allocating a user such privileges is done via well-defined processes with the acknowledgment of legal responsibilities and consequences. The implementation is based on the “least-privilege” concept. Such accounts are being monitored and controlled regularly, and in case detecting non-compliance or violation against the least-privilege concept, the account in question is limited or removed.

5.1.11 Strong Password Policy

When a user wants to set or reset his native login password, she is required to choose a strong password with a defined policy, enforcing a password with minimum-length AND usage of caps / non-caps AND special-characters.

5.1.12 Cryptography Application on Sensitive Data

In the case of handling sensitive data such as passwords or API access-keys (by persisting and fetching), usage of cryptographic algorithms and approaches are defined and required.

5.1.13 Encrypted Data at Rest

All the data persisted in storage systems of the infrastructure are encrypted at the disk level. This is valid for all the files stored in S3 Buckets and schemeless database systems.

5.1.14 Transport Layer Security (TLS)

Since MPP.360 is offered as an online software, the communication between the user’s internet browser and MPP.360 is encrypted by the HTTPS protocol. The certificate is issued and renewed automatically by Amazon (128 Bit Keys, TLS 1.3, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256). Also, intercommunication between the MPP.360 core and all other modules is secured with the TLS connection.

5.1.15 SQL Injection Prevention

Each single database transaction between the MPP.360 software and the database server has to be strictly sanitized and checked before being performed onto the database. I.e., the users or other actors cannot manipulate the SQL queries by entering malicious input. The prevention is done by the ORM placed between the software and the database engine. Additionally, there are WAF rules to detect and prevent obvious malicious SQL injections attacks.

5.1.16 Regular source code security scans

To maximize the security at the code level, and to make sure that the utilized software packages are all secure, a code scan with the tool “Snyk ™” has been set up. This procedure is done automatically, as soon as the master branch of a software module has been updated.

5.1.17 Software package integrity

All the software packages which have been used in all modules, are always downloaded and installed securely, directly from the packages’ repositories like Packagist or NPMJS. This process is done fully automatically by the standard tools like Composer and NPM (Yarn as well), to make sure that integrity of software packages has been kept and no manipulation / compromised code can be injected into the software. Besides the integrity aspect, all the software packages have been locked with a fixed stable version, so the compatibility of software components with each other is guaranteed.

5.1.18 Supply chain security

To keep conformity with the latest security regulations and recommendations, security requirements are communicated from the MPP.360 to the contractor, and all possible future contractors as a part of a clear-defined process. During this communication, it also has been made sure that the contractor(s) only get the least information possible (need-to-know principle) from the MPP.360.

5.1.19 Non-conformity detection and reaction

In case of detecting a non-conformity in the whole process of requirement engineering, software development and the end delivery, all the participants have been made aware to report such an incident to the relevant supervisors. By receiving such a report, corrective action to address the issue is planned and executed accordingly. Such cases are then used as a knowledge base, to reduce the chance of repeating the same incident again in future.

5.2 Infrastructure Security

It is also crucial to secure the infrastructure in which the software operates. The following describes how the underlying infrastructure of MPP.360 is secured.

5.2.1 Governance Model

To implement infrastructure security in an organizational and operational manner, AWS Control Tower and Security Hub are used. Security Hub scans and tests all infrastructure components according to the CIS Benchmarks Standards automatically and regularly. In case of detecting a vulnerability, the issue is reported alongside its severity and possible resolution. To maintain a high standard conformance, these security reports are reviewed constantly and in case a need for action has been detected, the necessary actions would be taken to resolve the issue. The procedure of finding and resolving security issues takes place for the underlying infrastructures of all MPP.360 modules.

Standard	Governance
CIS Benchmarks	• Identity and Access Management • Storage • Logging • Monitoring • Networking
AWS Foundational Security Best Practices	The whole infrastructure is also scanned regularly with. A list of covered resources by these security measurements can be found here
Control Tower	• Data Protection • Identity and Access Management • Compliance Validation • Resilience • Infrastructure Security

5.2.2 Datacenter Security

The physical location of datacenters on which the AWS services are consumed, is Europa (mainly Frankfurt/Germany). The infrastructure complies with the C5 (Cloud Computing Compliance Controls Catalogue) standard, covering the following qualifications:

- ISO/IEC 27001:2013
- CSA Cloud Controls Matrix 3.01
- AICPA Trust Service Principles Criteria 2014 (AICPA - American Institute of Certified Public Accountants)
- ANSSI Référentiel Secure Cloud 2.0 (Draft) (ANSSI - Agence nationale de la sécurité des systèmes d'information)
- IDW ERS FAIT 5 04.11.201 (draft of a statement on accounting: "Grundsätze ordnungsmäßiger Buchführung bei Auslagerung von rechnungslegungsrelevanten Dienstleistungen einschließlich Cloud Computing" ["Generally accepted accounting principles for the outsourcing of accounting-related services including cloud computing"], 4 November, 2014 Version)
- BSI IT-Grundschutz Catalogues, 14th Version 2014
- BSI SaaS Sicherheitsprofile 2014 [BSI SaaS Security Profiles 2014]

More information can be found under: https://aws.amazon.com/compliance/bsi-c5/

5.2.3 Web Application Firewall

All the public endpoints of the MPP.360 to which a user or another system can send a request are protected by a Web Application Firewall in place. This applies to the whole frontend, as well as the API endpoints. By default, the firewall is configured to block requests originating from specific sources, and requests which are suspicious to perform malfunctions like SQL-Injections are blocked.

Upon the request of the tenant, it is also possible to configure the firewall further with rules only applicable to the client's environment.

5.2.4 DDOS Attacks

After a request passes the Web Application Firewall, it is processed by an Application Load Balancer which is protected against DDOS attacks.

5.2.5 Load Balancer Protection

The Application Load Balancer itself is protected by security group rules which only allow HTTP traffic on port 80 and 443. Connections to other ports are refused. It is noteworthy that the traffic on port 80 is being redirected immediately to port 443, in order to apply TLS on the request.

5.2.6 Worker Nodes Security

Each server instance located behind the load balancer is configured to not accept SSH keys at any given time. Besides, the worker nodes are configured to have no public IP addresses and none of them are reachable over the internet.

5.2.7 Network Security

To prevent access to the operational servers from the internet, different private subnets have been set up. All operational servers, regardless of their type, reside in private subnets and access to these resources has already been prevented at the network level. Besides this, all the resources are protected from incoming network directly, and their outgoing connections have been routed through managed gateways.

5.2.8 Database Security

The database cluster on which the database of MPP.360 is hosted is protected by different security levels:

- The nodes are only accessible through the application servers and have no direct connection to the internet.
- In the case of technical issues and maintenance, only authorized staff have access to the database instances management console in a temporary basis.

5.2.9 HTTP Endpoint Security

SSL connections are always enforced using HSTS.

5.3 Information Security

Regular and active information security training and webinars are held by the contractor with respect to software development and infrastructure security. Beside this, all the relevant employees of the contractor have to do regular information security and data-protection (Datenschutz) online training and certifications in the course of the project. Technical Leadership makes sure that such trainings and certifications are taking place regularly and successfully.

5.4 Access Revoke upon Team Structure Change

In the event of team restructure, both in MPP.360 and contractor's environments, all the existing access and privileges which have been granted to the team-members in question, are re-evaluated and possibly revoked.

5.5 Sole Responsibility as Asset-Ownership

For each single digital and non-digital asset which is relevant to the project, there exists a dedicated sole responsible person as the Asset owner. Every grant and revoke of access to the assets in question is done via an explicit review of the asset owner. This holds for MPP.360 and the contractor.

5.6 Systematic Credentials and Access Management

For every single transaction regarding credentials and access keys, there exists a defined process and implemented by standard access-management tools. This is valid for MPP.360 and the contractor.

5.7 Regular Pen-Test Routines

In addition to best security practices at the design and implementation level, a Pen-Test procedure is scheduled every 12 months. To prevent influence and bias, the Pen-Test is conducted by an individual security group which doesn't have regular contact with the development team. Of course, to address and correct the findings of a Pen-Test, an active communication link between the Pen-Testing group and the developers is established. The Pen-Test routine includes the whole solution (Software and Infrastructure) and makes sure that the solution keeps the latest security practices and recommendations. The results of every Pen-Test routine, findings and their corrections are documented.

5.8 Regular Security Patches

Since all the software and infrastructure components have been monitored for security findings constantly, regular security patches are being applied constantly as well. This is done in different levels:

- AWS Patch Manager for ECS/EC2 clusters
- Security patches as software upgrades inside the docker containers
- Security patches as software upgrades inside the lambda layers

5.9 Physical Security of Workplaces

At both MPP.360 and the contractor's offices, physical security as well as policies like the clean desk policy have been defined and implemented.

5.10 End-Device and Resource Access Security

By the contractor, all the digital devices used for development and maintaining MPP.360 software solution, are secured both physically and internally by software measurements like encrypted disk drives, multi-factor authentication by login, two step verification when accessing software development resources, regular virus and malware scans of desktop and E-Mails, with the updated virus and malware scanners. Also, internal policies like locking screens upon leaving the device, clean desk policy, etc. have been defined and enforced. Besides these, accessing sensitive software resources from outside of the offices is only possible via a secure connection to the internal VPN of the contractor.

6 Data Integrity

6.1 Transaction Demarcation

MPP.360 persists application data using an abstract ORM layer. All write operations are first queued until a final “flush” command is issued. At this point, the ORM takes care of committing/rolling back a transaction.

6.2 Concurrent Data Processing

Since it is possible for two users to open the same job and work concurrently, MPP.360 explicitly notifies all the users that have the same job open with a prominent warning. This is a precautionary measure to prevent incidental updates from different users.

6.3 Multi-Tenancy

MPP.360 is a multi-Tenant system. The data of all tenants is saved in the same physical database environment and is logically isolated using an MPP.360 internal client ID.

Also, for saving user uploaded files a single S3 bucket is used and the files from different tenants are separated logically by using different S3 keys.

6.4 Encryption

Jobs’ textual data can be saved encrypted (on application level) with tenant-specific encryption keys. This feature must be activated for each client through the SysAdmin panel. By doing so, individual encryption keys are generated for the client. Then the administrator must configure a pattern explicitly, which fields must be saved encrypted. For encryption / decryption the AES-256-CBC algorithm is used.

6.5 Outsourcing Job-Data

It is also possible to store the jobs’ data in an external source, operated by a microservice which has to implement a given data interface. This feature has to be configured for each pattern individually. By doing so, no job data would be saved on an MPP.360 database. This feature can be combined with job-data-encryption, too.

6.6 Data Integrity at Transfer and Reception

All the data transfer between the software system and third-party systems, as well as internal communication between system components are implemented with TLS secure connection, keeping the data integrity between the sender and reception during communication.

7 Business Continuity

Keeping the organization functional during technical difficulties is another aspect that MPP.360 addresses extensively. The following explains briefly how the MPP.360 detects, plans and responds to technical incidents.

7.1 Governance & Monitoring

To monitor the internal behavior and the failures that might happen at the software level, MPP.360 is being monitored and tested automatically using DataDog ™ (European Instance).

Every erroneous action in all available environments (Production, Staging and Testing), with the root cause of the error can be detected and fixed as quickly as possible.
Also, the essential functionalities of the software (Login, start a new job, run workflow steps, logout) have been setup to be tested automatically every hour. Therefore, the availability of the core components (Application, Database, File Storage, Workflow Engine, E-Mails) is assured. If a shortage in a key component or service is detected, technical staff and the PO are informed immediately.

7.1.1 Logical Failures

Besides the service availability MPP.360 also reports logical failures during its runtime if detected. Logical failures are referred to bugs, or unexpected conditions that might happen inside the software while all services and components were up and running. Therefore, a high level of QA is achieved during the runtime operation.

7.2 Backups & Recovery

To provide a continuedly high level of business continuity, MPP.360s backup concept covers all relevant types of data: Configuration, Customer data & -Uploads, Applications, and Infrastructure.

7.2.1 Point-in-Time recovery

For Databases and File Storage, Point-in-Time recovery is enabled for all sub-accounts. It provides a manual and selective rollback up to 35 days to a certain state of a services. The service is “built-in” and managed by AWS natively and applies to databases and file storage (s3).

7.2.2 Database Snapshots

Automated backups of MPP.360 core database take place every day between 2:00 – 2:10AM (CET) by RDS Automated Snapshots. Database snapshots of the previous 35 days are available to use, and older snapshots are deleted automatically.

7.2.3 Separate AWS account backup for each module

Using AWS standard backup service, all sub-accounts including all services are backed-up into dedicated vaults residing in the same AWS account. Automated backup monitoring, schedules and retention management is activated and in use.

7.2.4 Source Code

Source code (application & infrastructure) is stored and maintained in Bitbucked. Backups are provided by the application itself.

7.3 Disaster Recovery

Regularly “System Recovery Practice” by the technical team is practiced. This ensures that backup processes as described above are handled correctly and a system restore can be executed at any time. In these practices, different aspects of the system like roll backs, restoring database backups or trouble shootings are practiced.

7.4 Scalability

Since MPP.360 and the modules are deployed as stateless docker containers using ECS or implemented as microservices using Lambda functions, scalability is part of the underlying architecture per default.

7.5 Availability

7.5.1 Availability goal

The defined goal is to have a minimum availability of 98% during a month. This includes the availability and usability of essential functionalities of the system, from an end-user perspective. Based on our experience and measurements a 99.9% system-availability has been reached.

7.5.2 Multiple Datacenter (Availability Zones)

As a cloud-based solution, MPP.360 Core uses a multiple datacenter architecture, having 3 individual data centers mainly located in Frankfurt, Germany. Different server nodes are spread across separated physical datacenters (AZ). Every server inside an AZ, holds multiple instances of the latest software version. Therefore, the availability of the software across different levels is guaranteed:

- Across multiple servers in one AZ: During an outage of a single server, other servers containing the application will cover for the failed node.
- Across multiple AZs: During a datacenter outage, other datacenters containing the application will cover for the failed AZ.

This architecture has been adopted and implemented for the application nodes, as well as the database cluster. It is noteworthy that MPP.360 is being hosted on a 3-AZ architecture.

7.5.3 Automatic Recovery of Failed Instances

Using AWS ECS, in case a server-instance or docker-container fails, it would be replaced automatically with another healthy instance.

7.6 Event-Logging & Auditing

To support transparency and have a shortest-time reaction to technical incidents, software and system events are being logged securely with various logging mechanisms. The table below, depicts an overview of these logs:

Type	Purpose	Access	User Data	Immutable	Remarks
Audit Log	Transparency, protocolling user interactions with the software	Secured, authorized user in software, MFA possible	NO	YES	Implemented as software feature

Connectivity Log	Detection and reaction to logical failures	Secured, authorized user in software, MFA possible	NO	YES	Implemented as software feature
Software Monitoring	Failure detection / performance Analysis	Secured, authorized admins, MFA	NO	YES	DataDog ™ is used, European instance
Container Logs	Failure detection and analysis	Secured, authorized users, MFA	NO	YES	Stored in CloudWatch Logs of AWS

Remark: MPP360 does not write PII data in any logs natively. However, all parameters configured by integrators may contain PII data and hence will be stored in application logs.

8 Interoperability

To provide a high degree of integration into third-party software like client's existing systems, MPP.360 is equipped with 2-way communication components. These are:

1. MPP.360 RESTful API: which is responsible for receiving inbound connections from a third-party software – this component can nearly do every single action that a normal user at frontend can.
2. MPP.360 Webhooks: which can initiate an outbound communication from MPP.360 to a third-party software.

Having the possibility of inbound and outbound communication with other systems, MPP.360 can be integrated deeply into existing system environments.