FINESTGOV

MOBILE WORKFORCE & ASSET MANAGEMENT

ARCHITECTURE & SECURITY

Abstract: This document provides technical information on the FinestGov software architecture, software components, applications and the implemented security mechanisms.

PAGE OF **CONTENTS**

APPLICATIONS

ANDROID
GOOGLE PLAY

IOS
APPLE STORE

WINDOWS 10
MICROSOFT-
STORE

WINDOWS 7
SETUP.EXE

CLIENT

PHONE
LAYOUT

TABLET
LAYOUT

HIGH /LOW
DPI MONITOR

TABLET/
PHONE
EMULATION

WEB
MANAGEMENT

CONSUMER
PORTAL

SERVER

CLUSTERNODE US

CLUSTERNODE UK

MYSQL DATABASE

MYSQL DATABASE

RESTAPI

GWT RPC

RESTAPI

GWT RPC

FINESTGOV
APPLICATIONS

ANDROID / IOS / WINDOWS 10 / WINDOWS 7

- Built with Apache Cordova to support multiple platforms
- Includes all necessary Cordova plugins, but does not include any application code
- Accessing the server URL to download application code
- Server URL contains version number, so multiple versions are supported simultaneously
- Windows/PC versions do not support location tracking, geo-fencing and beacons
- Smart location tracking preserving battery life by tracking device motion

ANDROID / IOS / WINDOWS APPLICATION FEATURES

- Camera and media – capture photo, video, audio
- Barcode – detect visit /package delivery on barcode scan
- Local and push notifications – new visit / task notifications, closing hours, etc.
- Speech to text and Text to speech – hands free operation, text dictation
- Panic button – create event, initiate phone call and start video capture

ANDROID / IOS APPLICATION FEATURES

- Location tracking
- Geo-fencing – automate visits / events based on location
- Beacons – automate visits / events on iBeacon or Eddystone transmission
- NFC – Near Field Communication – detect visit on tag scan
- Touch ID / Fingerprint Scanner – log in with fingerprint authentication
- Navigate to address / geo location – use navigation apps installed on the device

FINESTGOV CLIENT

- Built with Google Web Toolkit (GWT)
- Contains local database (SQLite) for offline support of the data
- The data is synchronized automatically. When offline, data is stored locally and synchronized when going back online. All operations are performed against the local database, unless historical data is requested
- Uses web cache manifest for offline support of the application code
- New version is checked on the server using the manifest on each start and once in a while during use. New version is downloaded and installed automatically with no user interaction
- Uses Google maps to display geo-based data (only works when online)
- Responsive UI for all screen resolutions, large screen activities implementation
- Vector based icons for any DPI

FINESTGOV
CLIENT FEATURES

- **Route –** frequently performed routine with a list of locations. Each location can be visited one or more times during the route. Routes can be scheduled from once a day up to once a month. Visiting a location may include a visit form as well as reporting specific events or exceptions
- **Task –** single request to perform something in general or in a specific location
- **Inspection –** one or more operations to be performed at predefined time intervals. Operations include filling forms and scanning documents. Inspections can be performed on various locations as well as employees and equipment
- **Event –** manually or automatically performed action. May contain a single photo or form. Can be reported in many places, including visit, route, inspection, location, equipment or in general
- **Form –** a list of questions of various types that can or must be answered to provide a valid report. Used in visits, events, inspections and tasks
- **Process –** a list of steps required to handle an event. Each process step may create a task, all together forming a workflow
- **Pickup & Delivery –** loading packages from warehouse, deliver and collect packages during visits, return undelivered and collected packages to warehouse to transfer to another employee
- **Document management –** display document attachments and capture photos of document pages to generate a multi-page PDF document
- **Time management –** clock in, clock out, show and update employee timesheets
- **Manage –** locations, employees, equipment – update information and create events
- **Reports –** generate and display reports

Numerous other tools & features.

Web Management Application

Easily understandable & customizable

- Built with Google Web Toolkit (GWT)
- Runs on any current browser, including IE
- Allows managing data on company or department level
- Provides generic access to all objects and object dependencies
- Supports displaying data as list, tree, object, document, map or calendar
- Menu permissions to limit access to features
- Object permissions to limit access to objects with read/write/create/delete access
- Department settings to force specific objects to be managed on company or department level or allows the user to specify on demand

Web Management Application Features

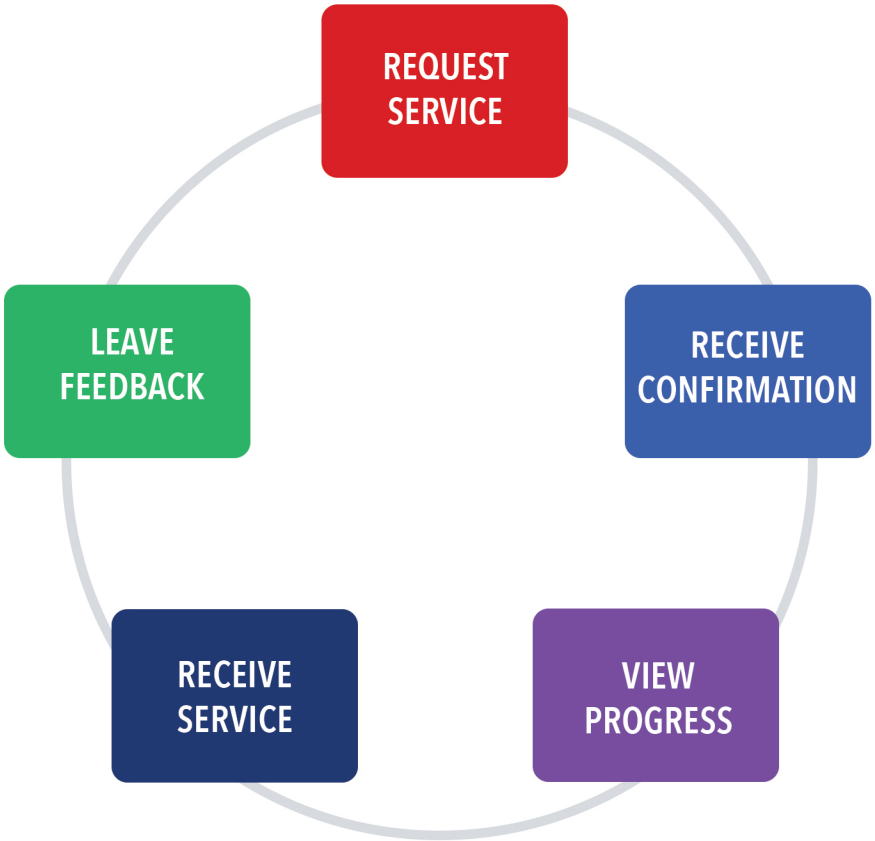
- Create and manage locations, routes, inspections, tasks, etc.
- Manage system settings, update all tables, manage types
- Schedule routes and visits
- Schedule pickup and delivery
- Create and manage tasks
- Design forms
- Review all data reported from the Client Application
- Audit all changes made at any time
- Location tracking – view employees on the map and track their movements
- Generate reports on demand
- Schedule reports to be generated and emailed on predefined times
- View all reports scheduled and emailed in the past
- Create client portal – shared reports that can run by third parties
- Design and view live dashboards with the essential information
- Manage panic buttons
- Design Client Application main menu
- Create consumer portal – access to the account from the Consumer Portal Application
- Design external interfaces (integrations), map tables and fields

CONSUMER PORTAL APPLICATION

- Maintained as a web based or mobile (Android / iOS) application
- Supports any modern browser (including IE) and mobile phone
- Standalone, online application with its own user interface and server
- Can be hosted on our or customer's servers
- Contains all Client Application functionality on the user's side
- Communicates with the Server using APIs, no direct access to the company data
- Each portal requires its own (sub)domain name or a specific mobile application
- Mobile applications designed for the consumers differ in branding and the domain name
- Allowing specifying which data is visible for consumers, by default no data is accessible

CONSUMER PORTAL
APPLICATION FEATURES

- Custom branding – display custom caption, web/application title and logo
- Login with email and/or phone by receiving a confirmation code via email or SMS
- Display a custom designed main menu
- Report events – initiate call, request service
- Fill forms to report events or interact with the service provider
- Display previously reported events
- Interact with the service provider on the open calls (messages)
- Show specific processes (steps) of the reported events – progress
- Receive (fully customized) notifications on the progress of the call
- Allow choosing location, equipment, list records and employee out of the allowed list
- Leave feedback and close calls



SERVER **ARCHITECTURE**

- Built with Java 1.8
- Uses MySQL 5.7 database
- Runs on Tomcat 8
- Installed on Ubuntu 18.04 server with LXD/LXC containers
- Database installed on the server
- Applications installed in LXC container
- Nginx used as a reversed proxy for Tomcat (for better security)
- Uses SSL only (HTTP redirects to HTTPS)
- Supports HTTP/2, Security Grade A+ https://www.ssllabs.com/

CLOUD INFRASTRUCTURE

- Servers installed in Microsoft Azure
- Currently, there are two servers in US East and UK South locations
- Traffic Manager (DNS based) load balancer routes traffic to the closest available server
- MySQL is implementing Master-Master replication between the servers
- Users can (potentially) instantly switch between the servers without losing any data
- No session synchronization is required
- Local cache (Guava) is used for performance improvement, cache invalidation is broadcasted to all servers on data changes
- File attachments (documents, photos, videos, etc.) are saved in Microsoft Azure blobs
- Logging and tracing are performed with Microsoft Azure Application Insights. Metrics collected for server load, request rate and timing, SQL calls, reports, notifications, emails and text messages, storage, API calls, etc.
- Application Insights generate alerts on any abnormal behavior such as high exception rate, slow operation or server failure

ON-PREMISE INSTALLATION

- It is possible to provide installation on the customers' server(s)
- Minimum requirements – single server 2 CPU 16GB RAM running Ubuntu 18.04
- Requires custom mobile applications to work with the custom servers
- File attachments can be stored in the file system, in Microsoft Azure blobs or Amazon S3

API AND SDK

- Implementing RESTful API for everything in the system
- Supports both XML and JSON for data in and out (consume, produce)
- Using Swagger to expose and describe the APIs
- SDK is provided in Java, JavaScript, .NET, PowerShell
- SDK provides simple, synchronized methods and full object modelling
- Can be used in integrations, BI, connected Excel (Power Query), web forms, etc.

FINESTGOV DATABASE

- Using MySQL 5.7 database on each server
- Master-master replication between multiple servers with GTID
- Multi-tenant environment, single database for all accounts, CompanyID field in all tables
- Main database with all tables has no permissions at all to any of the users
- Another database with no data tables is the only visible database
- The secondary database contains views listing only data allowed to the current user's company and department
- The secondary database contains stored procedures, retrieving data from the views
- The application is using stored procedures only, no direct table/view access is allowed
- Two layers of security – views are limiting the allowed data; stored procedures are

implementing the logic to retrieve the right data. A bug in the application or stored procedure will never expose any data from other departments or companies

- A third database is used for user authentication and is not visible to any of the users. The Users database contains several stored procedures that are called by the secondary database to add/update users and authenticate users logging in
- The application connects to the database with a single user. After connection, no data is available (all views are empty) until logging in with username / password or token

USER AUTHENTICATION

- Company users are created and managed in the Web Management application
- The first, administrative user is created when creating new account
- The users are saved in the Database
- First time, users log in with their username and password using Client or Web Management Application
- Applications call a stored procedure in the Database, providing the username and the password's hash (SHA1). Passwords themselves are not saved, but only their hashes
- If the provided username and password matches, a token is generated and saved in the database. The token is wrapped with JWT (with expiration) and returned to the client
- Tokens are random and created using several UUIDs
- All next calls from the client are performed with the token
- Applications provide the token to the Database on each connect. The token is used in the views to allow any data relevant to the user
- New token is generated on every login. Old tokens expire on their own
- It is possible to generate a permanent token (with no expiration) for using in APIs, scheduled reports, BI, connected Excels, etc.
- It is possible to manually invalidate any existing tokens, including permanent ones

FINESTGOV
INTEGRATION

- All data can be retrieved from the system and saved as Excel files using the Web Management Application
- All data can be retrieved as JSON or XML using a single API call (for each object type)
- All data can be retrieved as an Excel file using a single API call
- Simple data import into the system using Excel or JSON files
- Mapping Excel files and/or sheets to objects and columns to fields
- Writing JavaScript code for complicated column to field mapping with logic
- After mapping all fields and manually importing the Excel (or JSON) file, automate the process with a single API call (using the existing mapping)
- Switch between Excel and JSON without remapping fields. Allows to use Excel for the first time and later automate with JSON
- Importing data into the system is also possible using API / SDK on a single object level by creating and updating objects
- Connected Excel allows connecting Excel to the system and automatically updating the data in the Excel. Custom logic can be implemented using Power Query M language

SECURITY

- Web applications installed in a container with no direct access to the data
- Database connection using the default user does not provide any data and does not allow any operations before authentication using username and password or token
- Local databases on mobile devices contain data relevant to that user only
- All connections between all applications and the server are encrypted by HTTPS
- Remember me option in the Web Management Application to allow or prevent token authentication after closing the browser window
- Remotely delete application data and kick out users from the Web Management Application
- Enable / disable users without deleting them
- User permissions to use Web Management Application, Client Application, change settings, manage other users
- Object level permissions – read, update, create and delete per employee group
- Application permissions to enable / disable functionality per employee group
- Account security settings provide token expiration time (the number of days users stay logged in without using the application)
- Account security settings provide complex password settings (minimum symbols, using letters and digits, uppercase and lowercase, symbols)
- Users cannot use the same password used in the past (remember last 10 passwords)