

# QiTASC

## PRODUCT SUITE

Contact us:

[www.qitasc.com](http://www.qitasc.com)  
[info@qitasc.com](mailto:info@qitasc.com)  
+43 1 810 21 73

Diefenbachgasse 53  
1150 Vienna  
Austria

  
the magic of E2E testing

## Products & Services

### intaQt® & Complemented Software

In addition to our flagship product intaQt® we offer a range of other products that complement one another in order to optimize the automation pipe.

#### Features

intaQt® offers a range of features that allow the user to execute and validate automated end-to-end tests based on complex use cases spanning multiple domains.

intaQt® supports all domains

Voice: 2G, 3G, 4G, 5G, POTS, ISDN, WiFi, VoLTE aut.

SMS, MMS, USSD automation

Data usage automation

Automation of provisioning & charging verification

Web-UI & app automation

IOT & 5G automation

CPE & device automation

### Workshop & Training

Workshops take place in our new training rooms on a regular basis to provide and maintain intaQt® knowledge. Training modules support and improve intaQt®-enhanced software development results.

### Managed Services

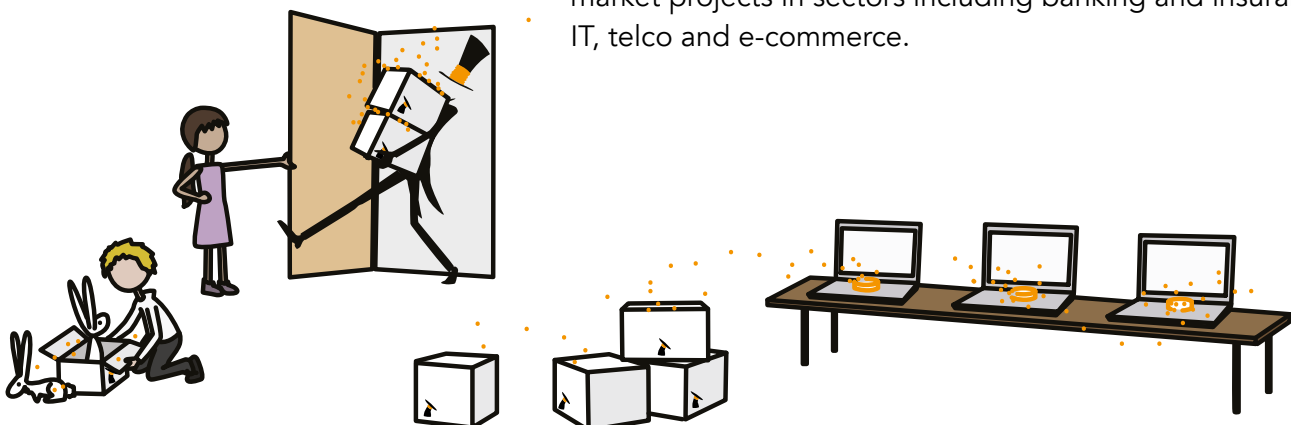
QiTASC provides licenses, a testing team, and project management to support customers' integration, provide configuration activities, develop and execute test cases (including verification), and carry out reporting and defect management.

### All-In Services

QiTASC focuses on test-driven development and continuous integration to satisfy project development needs.

### Business Partnering

International partners resell QiTASC's products for local market projects in sectors including banking and insurance, IT, telco and e-commerce.

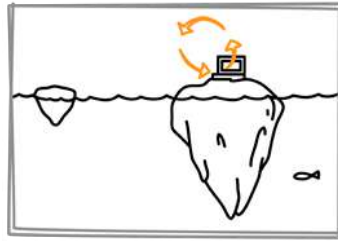


intaQt® covers the testing scenarios you are looking for!



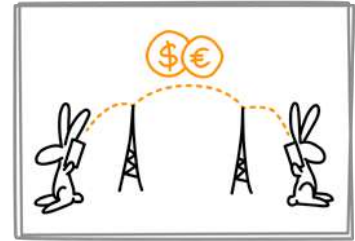
### Acceptance Testing

The the software itself to ensure that your system is running smoothly.



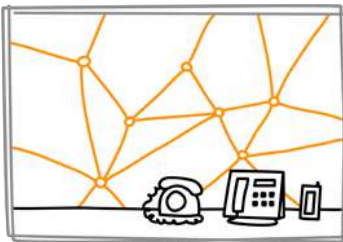
### Backend Testing

Verify server-side functionality and the communication between interdependent components.



### Tariff & Charging Verification

Make sure that convergent charging systems, online charging systems, and customer accounts are behaving correctly.



### Core Network Testing

Confirm that all aspects of a network behave correctly for telephony, SMS, data, and everything else behind the scenes.



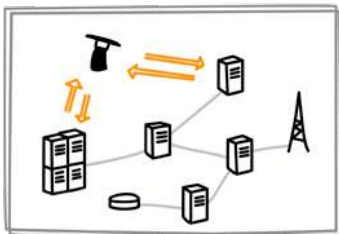
### E2E & Migration Testing

Check that a network behaves exactly as it would when triggered by a customer, even for time-sensitive projects like migrations.



### IFRS Testing

Ensure compliance with IFRS standards comparing huge sets of metadata against expected outcomes.



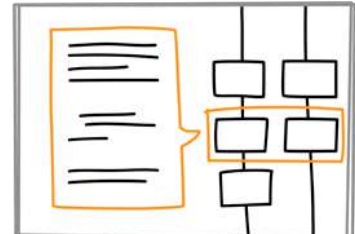
### NW Element Simulation

mimiQ is a more general-purpose "interface testing tool". It is a test script executor with built-in support for any kind of http communication.



### Internet of Things Testing

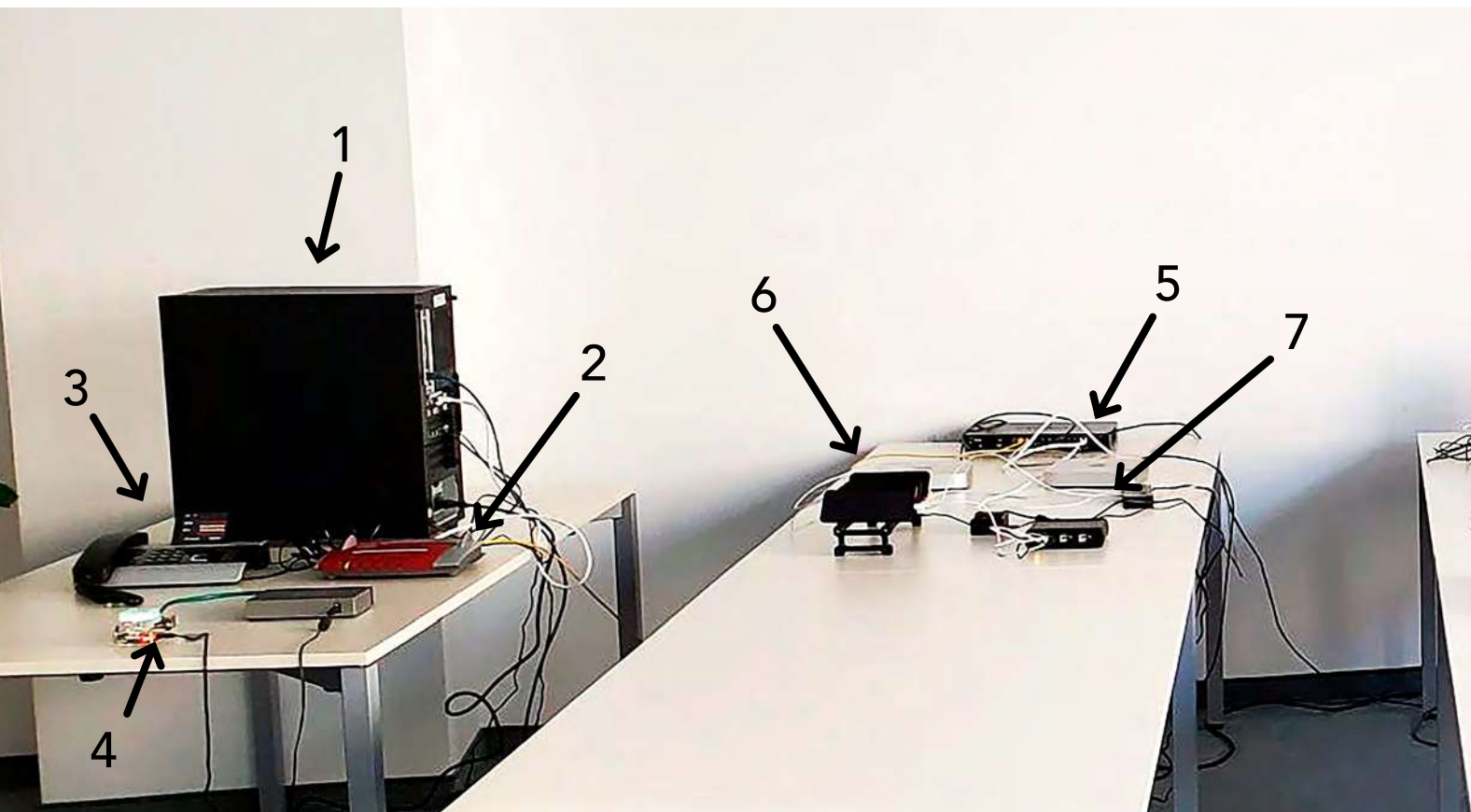
Automate smart devices to check that they communicate with their app and are working properly.



### Protocol Testing

Ensure that data is transmitted between devices and over networks according to protocol specification.

## QiTASC Lab:

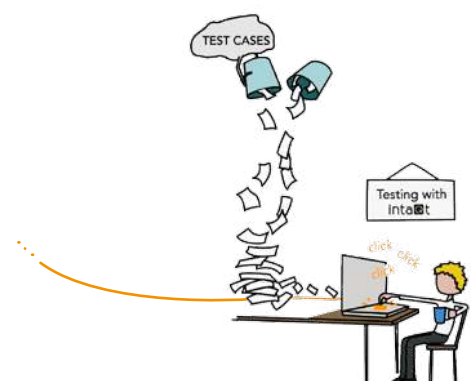


### Legend of the QiTASC lab:

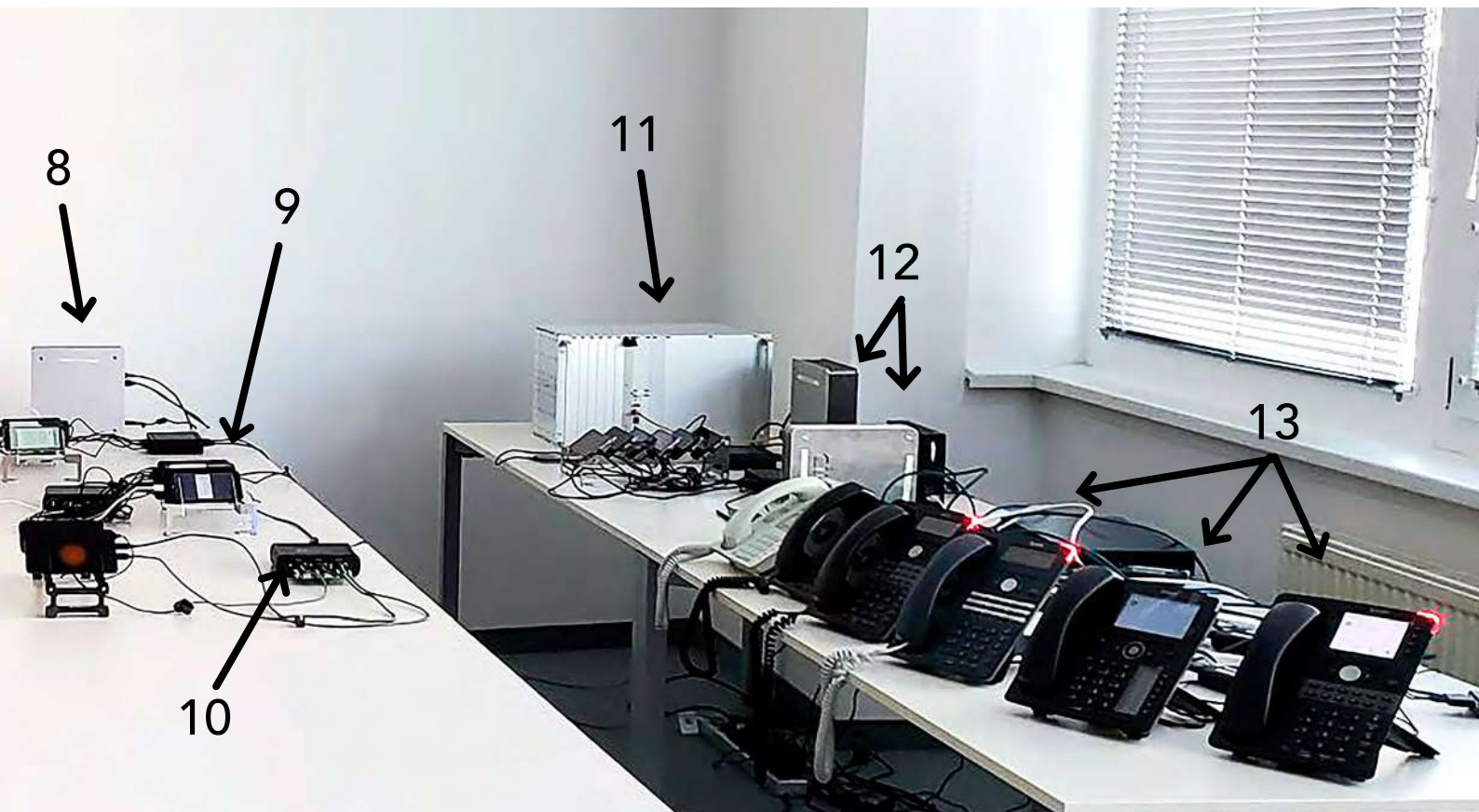
- 1 PSTN-IPS with Openvox card for CPE Testing
- 2 CPE Fritzbox 7490
- 3 Analog phone
- 4 RasPI IPS (mini IPS)
- 5 Mac mini IPS for iPhones
- 6 iPhones
- 7 phone-hub
- 8 Standard IPS
- 9 Android phones
- 10 phone-hub
- 11 SIM array (implementa)
- 12 Standard IPS
- 13 VoIP phones (Snom & Yealink)

### Legend of the IoT Setup:

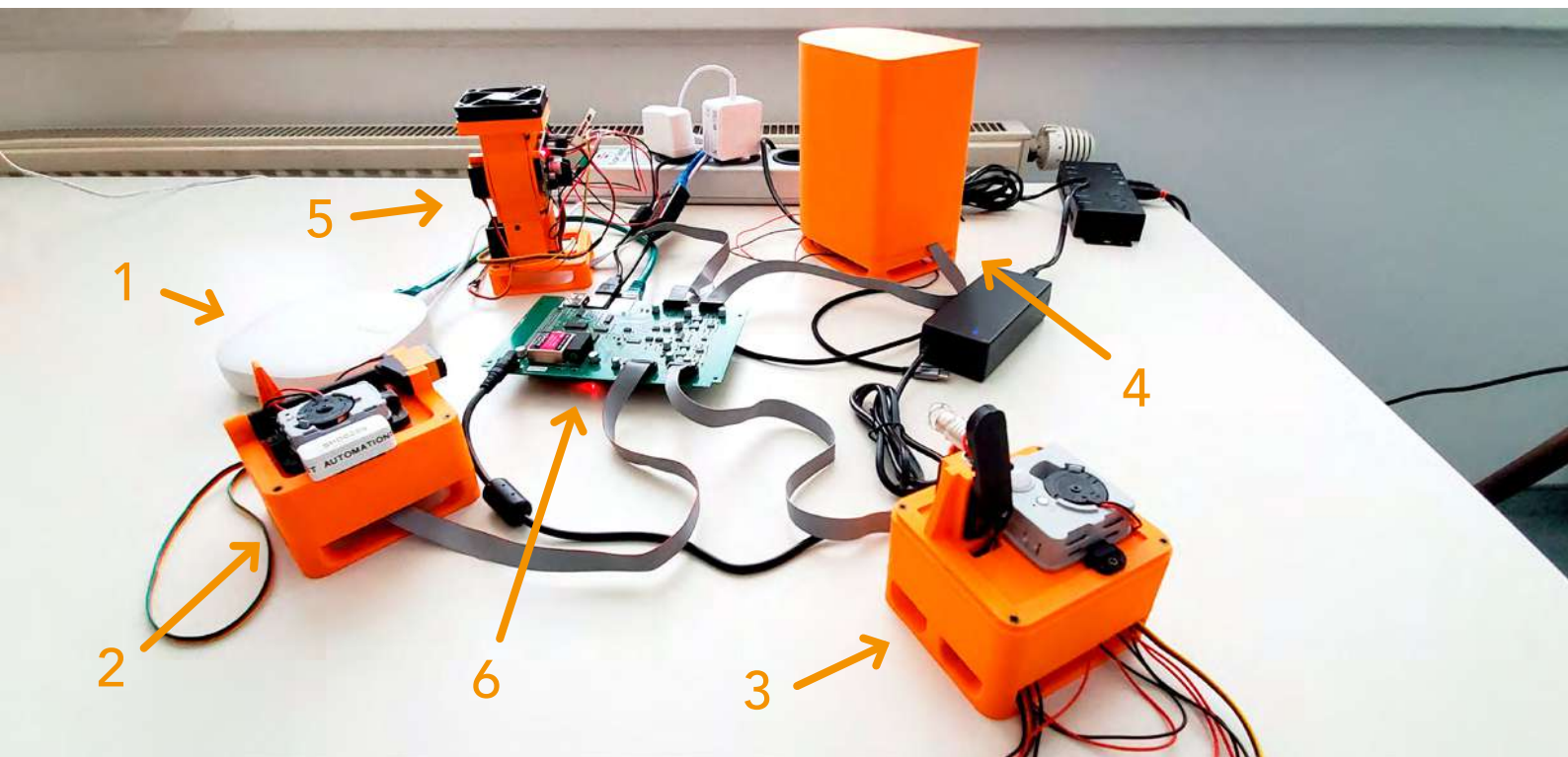
- 1 Hub
- 2 Door/Window Contact
- 3 Motion Detector
- 4 Bulp
- 5 Smoke Detector
- 6 Generic Platform Controller





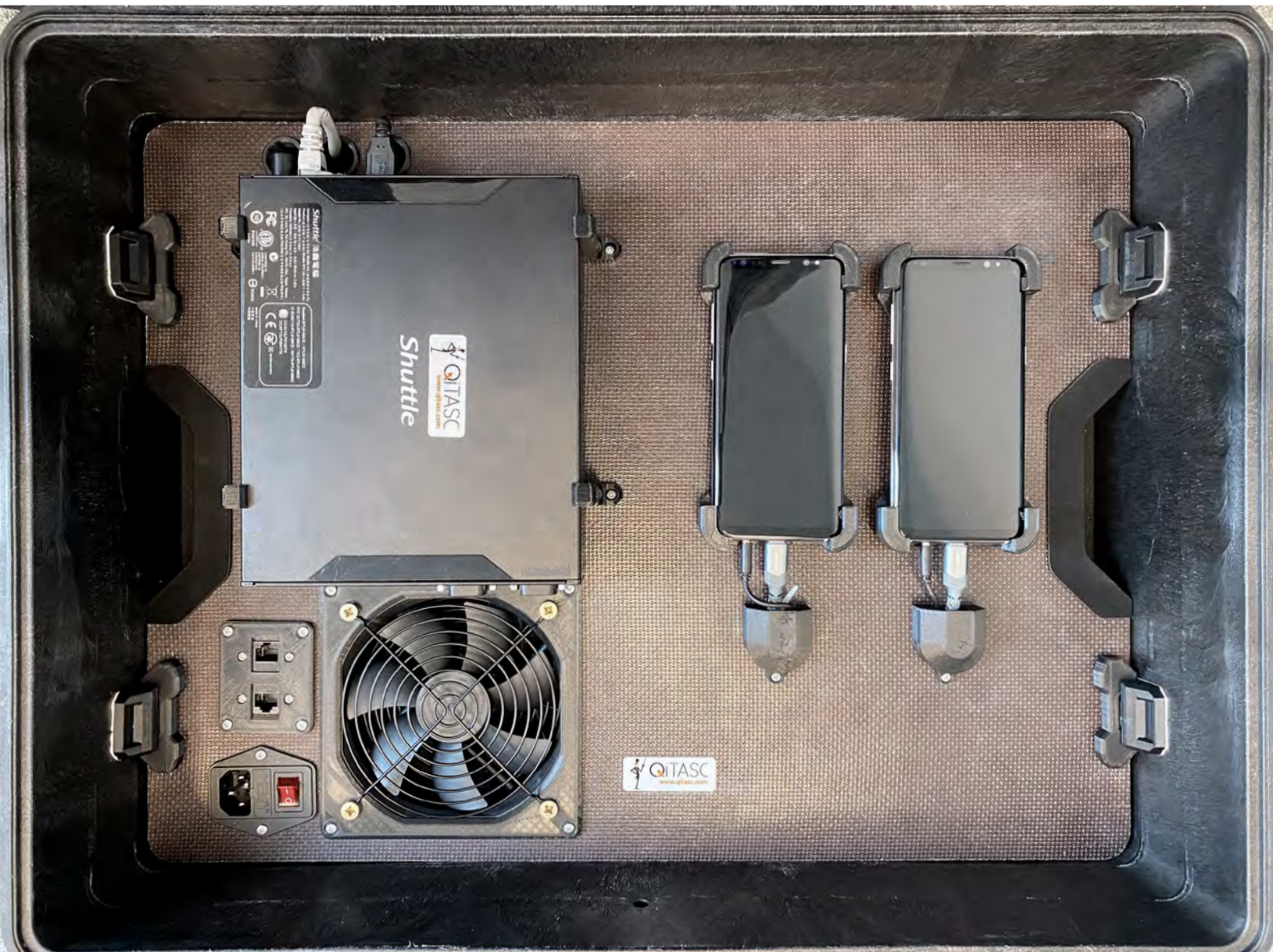


### IoT Setup:





## QiTASC Suitcase



The intaQ<sup>®</sup> Case has been designed as an out-of-the-box, plug-and-play production testing unit. It inherits:

- 2–5 phones
- an IPS
- phone-hub
- and fan.

All devices are already wired and can be used immediately. The systems are configured: insert the SIM cards, connect it to power and LAN >> GO!

## Product Overview

Get to know details of our products in the appendix.



### intaQt

#### Manage Test Automation Projects

Create, manage and execute automated tests that reflect real-world scenarios, customer experiences and complex interactions with backend systems. intaQt® supports real and simulated devices, and provides configurations to ensure that resources, including phones, are managed efficiently and that test cases and their results reflect real-life scenarios.



### intaQt studio

#### Test Case Development Environment

intaQt studio supports you and your code as you create test cases. Our integrated development environment, which is based on the IntelliJ IDEA platform, includes the most important and useful features required for robust and easy-to-understand test cases.



### intaQt client

#### Command Line Test Case Development

intaQt client provides a customizable way to execute test cases and projects, including XML configuration files, via the command line.



### intaQt verification

#### Verification

Verify hundreds of parameters and metadata of a test case within seconds during a test or after data has been collected. With the click of a button, intaQt verification checks device, subscriber and event details, internal infrastructure, legal requirements, financial characteristics and more.



### intaQt web-ui

#### Web-based User Interface

intaQt web-ui is a portal with a web-based user interface. It is a complementary product to intaQt studio, which offers the option for web-hosted test services.



## cheQ

### Trace Comparison & Verification

Compare two traces in an intelligent, automated way. Trace verification can be customized by packaging of trace data and correlation rules, simply edited in the tool (copy and paste). It also enables a record of previous comparisons to be kept.



## sQedule

### Resource-aware, Intelligent Scheduling Service

sQedule is a resource-aware smart scheduler that schedules tests on intaQ agents, then evaluates and allocates resources. It provides a monitor and actions to interfere with the test run.



## colleQtor

### Evidence Collector

colleQtor is a service for collecting evidence which comes as a stream and is then provided to the test case for verification purposes.



## conQlude

### Integrated Reporting and Issue Tracking

conQlude is a centrally accessible, secure web interface that collects, manages and exports intaQ test case project data and includes automated bug and error tracking.



## mimiQ

### Simulator for Network Elements

mimiQ lets you define https message sequences. It can act as a simulator for a network element in telecommunications networks, or provisioning systems. It is also operated as a load and stress test system, especially for 5G Nchf, Npcf, etc. interfaces.



## cdr-linQ

### CDR Web Service

The web service for collecting, filtering and presenting CDRs which have been generated by test cases. They are collected and decoded. Cdr-linQ shows two CDRs and compares them.



## reloQate

### SIM Card Detector

reloQate is a background service for checking and mapping SIM cards located in a SIM array/SIM Mux to a dedicated phone.



## subscriber-db

### Data Store

subscriber-DB is a data store for flexible SIM and device asset management, profile management and assignment, and much more.





# intaQt

## Manage Test Automation Projects

Create, manage and execute automated tests that reflect real-world scenarios, customer experiences and complex interactions with backend systems. intaQt supports real and simulated devices, and provides configurations to ensure that resources, including phones, are managed efficiently and that test cases and their results reflect real-life scenarios.

## Platforms and Languages

intaQt® is fully compatible with Windows, Linux and Mac OS, and includes a built-in vocable expression language as well as the custom QiTASC steps language and the QiTASC UI steps language for testing web and app functionality.



## Telephony

Our software helps introduce, test and integrate new services, technologies and legal requirements into telecommunications products. At the touch of a button, intaQt® executes complex end-to-end testing scenarios quickly, accurately and efficiently.



## Telephony - Audio Testing

The intaQt audio service enables the automated testing of audio recordings, audio matching and speech channel monitoring within telephony test cases. This is especially useful for verifying IVR prompts, audio recordings that subscribers hear or assessing sound quality.



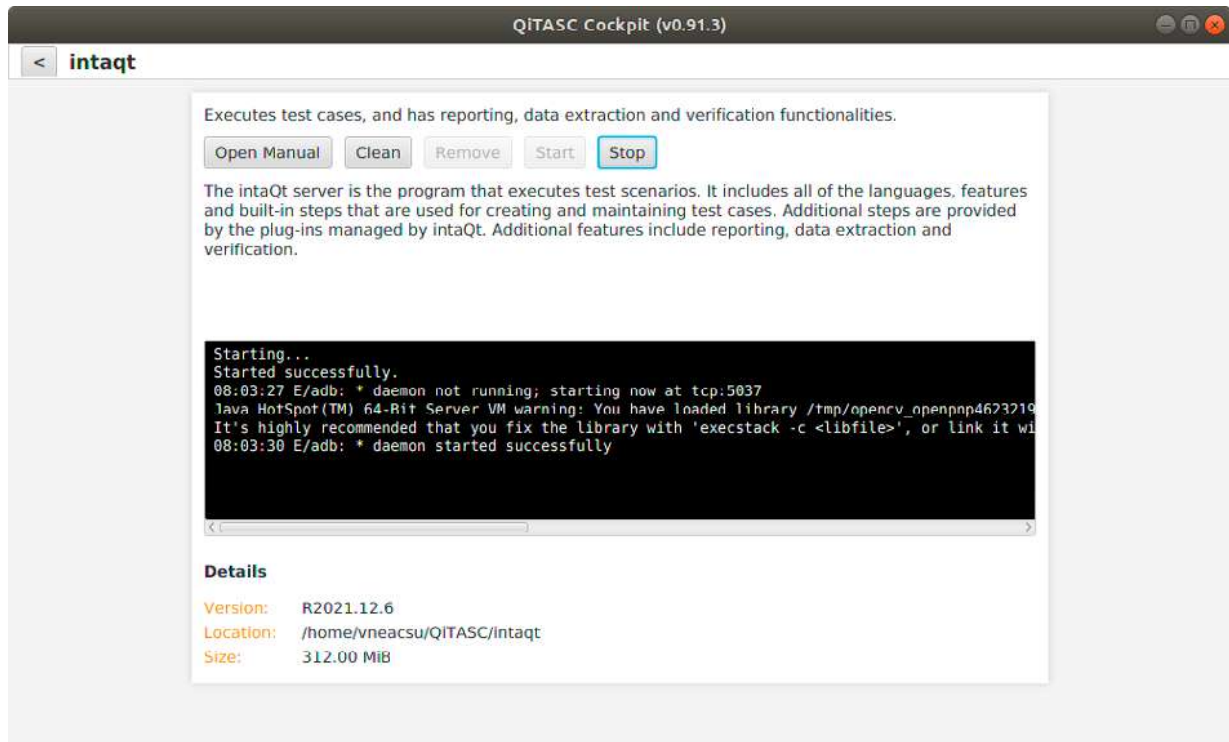
## Web, App and IoT

intaQt UI steps language automates activities performed while using websites, Android & iOS apps and Internet of Things (IoT) devices. This is done by defining elements, such as XPath, applying actions to these elements and writing test steps that mirror a user's real-life behavior.



## Built-ins for Backend Systems

intaQt® includes extensive built-in functionalities for communicating with backend systems. This includes HTTP/REST, SSH, JSON, XML, SQL, e-mail and more.



the QiTASC Cockpit allows users to manage all QiTASC products from a central place

```
1 stepdef "post string"
2   body := Http.stringEntity("Hello world", "UTF-8")
3
4   request := Http.post("http://httpbin.org/post")
5     .setEntity(body)
6     .addHeader("hello", "2")
7     .build()
8
9   response := Http.executeHttpRequest(request)
10
11   println(response.getContentString())
12
13 end
14
15
```

```
1 Feature: Basic Call
2
3 Scenario: Basic Call
4
5   Given a phone as A:
6     * of type Android
7     * where operator = "3 AT"
8     * with profile profileA
9   And a phone as B
10  And A starts a call to B as MYCALL:
11    * detect incoming call within 10 seconds
12
```

gerkhin

steps language

```
1 stepdef "open google page"
2   myView := open MyGoogle
3 end
4
5 stepdef "do google search for {}" / searchText /
6   GoogleModel.performSearch(searchText)
7 end
8
9
10
11 view MyGoogle
12
13   url "http://www.google.at"
14
15   elem searchField := "//*[@id='lst-ib']"
16   action (searchField) defineSearchText := type
17   action (searchField) clearSearchText := clear
18   action (searchField) doSearch := submit
19
20 end
21
```

UI steps language



# intaQt studio

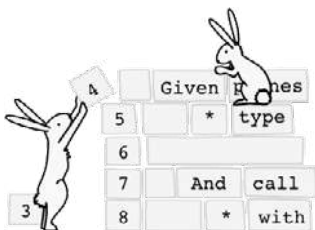
## An IDE Made for Testing

intaQt studio supports you and your code as you create test cases. Our integrated development environment, which is based on the IntelliJ IDEA platform, includes the most important and useful features required for robust and easy-to-understand test cases.

Beginners and experienced testers alike find that intaQt studio helps them write, execute and manage all aspects of their intaQt® test projects.

## Platforms and Integrated Language Support

intaQt studio is fully compatible with Windows, Linux and Mac OS. Our languages, steps language, UI steps language and vocable expression language, are supported by all the most important productivity features for writing and running test cases, including code completion, refactoring, and inspections.



## Code Completion

Intelligent code completion is an essential part of test case creation and development. intaQt studio supports code completion for both built-in and custom steps, steps and UI steps languages and configuration files.



## intaQt studio Phone Plugin

The new intaQt studio phone plugin app lets users access devices, find information about them, and watch phones work in real time. This means testers no longer need to leave their desk or contact someone at another location to check on a phone's

status or characteristics. The phone plugin lists all available phones in an active project, both local and remote. Manually selecting a phone displays its active screen during test executions, and allows toggling to view the phone's properties.

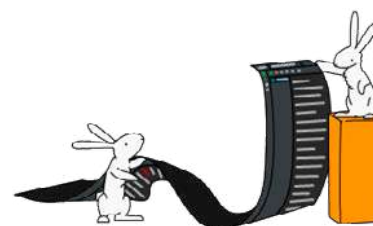


## Inspections

When intaQt studio determines that a function is invoked, it attempts to find the function definition in the respective object. If no matching signature is found, the function invocation is marked as an error.

## Productivity and Developer Tools

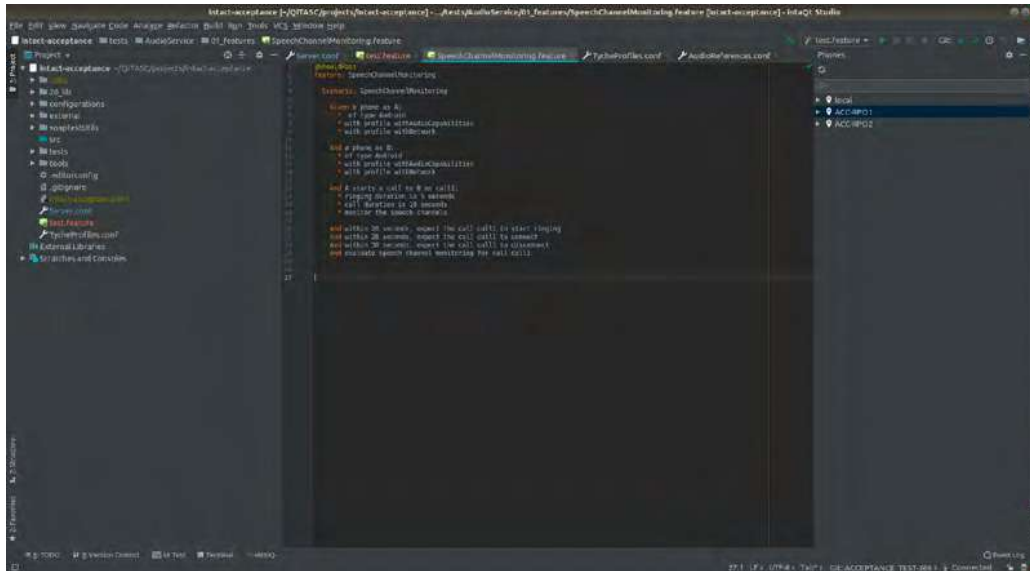
Additional productivity and shortcut tools simplify navigation through intaQt studio and let users integrate projects into version control systems and access command-line tools.



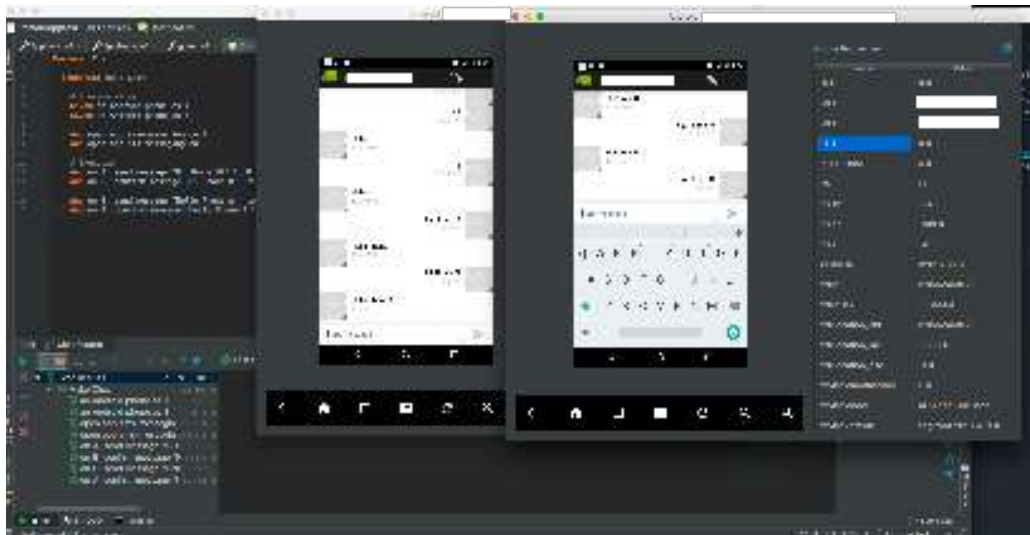
## Logs and Reports

Toggle between trace, protocol, and server logs.

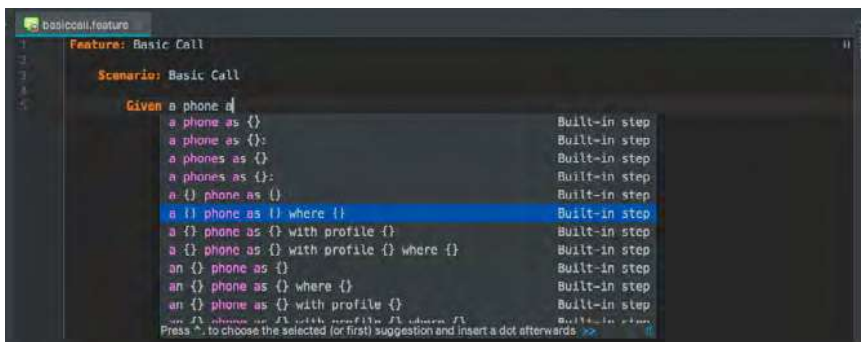




intaQt Studio is a convenient IDE for writing and executing intaQt® test cases. It provides features such as code suggestions, navigation, inspections and interactive control of remote intaQt® phones



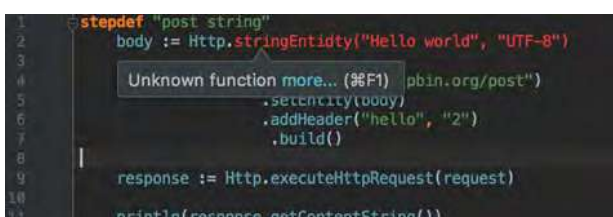
intaQt® with remote phones



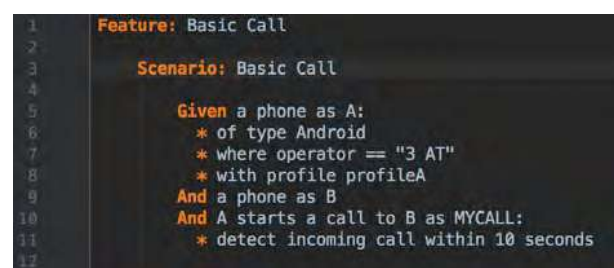
built-in steps



access device information



unknown functions and models



vocabulary expression language





# intaQt client

## Command-Line Interface & Continuous Integration Support

intaQt CLI is a command-line interface & continuous integration support and provides a customizable way to execute test cases and projects, including XML configuration files, via the command line.

Maximize your project's level of automation by executing entire intaQt® test projects in a continuous integration (CI) environment as part of a build. intaQt client provides a customizable way to execute test cases and projects, including XML configuration files, via the command line.



## Flexible Configurations

intaQt client enables configuration of all of the following to ensure entire workflows are handled automatically:

- Project configuration files, which take precedence over default configurations
- XML report directories
- Automatically create summary report files after test executions
- Synchronize and upload local project changes to the server before execution

## Cross-platform Command-Line Functionality

intaQt client is fully compatible with Windows, Linux and Mac OS. All the features and parameters below listed can be run from standard command-line interfaces.

## Test Case & Execution Criteria

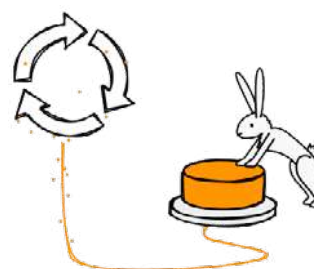
intaQt client provides parameters for test case selection and execution including:

- Tags for filtering test cases
- Number of retries for failing test cases
- Delays between each test case execution

## Network Settings

Specify where test projects are hosted and execute tests remotely.

- Host name on which the project should be executed
- Port number



## Continuous Integration

Configure test projects in Jenkins or TeamCity to run as a build.

In our online newsroom we offer a tutorial, in which we show you how to incorporate intaQt® into the Jenkins continuous integration framework. Try out our sample project out for yourself and see how easy it is to further automate your testing!



# intaQt verification

## Verify Complex Test Data

Verify hundreds of parameters and metadata within seconds during a test or after data has been collected. With the click of a button, intaQt verification checks device, subscriber and event details, internal infrastructure, legal requirements, financial characteristics and more.

## Platform and Language Support

intaQt verification is fully compatible with Windows, Linux and Mac OS. Its rules files are compatible with intaQt's vocable expression language and UI steps language, meaning that verification test cases, their steps, and configurations can be integrated into larger intaQt® projects.

## Verification Rules

intaQt verification automates the comparison of call detail records (CDRs) or event detail records (EDRs) and XML files against a configured set of verification rules. Rules define the type of check that intaQt verification will perform against the selected records, as well as which properties to compare. Its rules check parameters regarding test case structure, comparisons, and formats. Examples of verification rules include:

- Test case structure rules - concern the structure of test cases or their tickets
- Comparison rules - compare ticket properties relative to specified criteria
- Format Rules - check values, which must have a specific format

## Verification Selectors

intaQt verification simplifies defining the criteria to be checked via its selectors. These selectors tell intaQt® which records to check against verification rules and which to exclude. Because only certain records or tickets should be chosen, selectors are a critical part of eliminating unnecessary data from the verification process, which helps to keep these

tests running smoothly and quickly.

## Online and Offline Verification

intaQt verification is available for both online and offline testing. Online verification involves live intaQt test cases, which execute verification while the test case is in process.

Offline verification happens after a test case execution is no longer in process. For example, if CDRs/EDRs are only generated once per day, online verification may not be possible. While it is possible to set up a project to use online verification in this case, the process is intricate and requires extensive know-how. Therefore, we have implemented a verification UI specifically for such offline use cases.

## Verification Built-ins

Verification built-ins provide additional flexibility and specificity in defining rules. These built-ins contain functions that filter for important criteria.

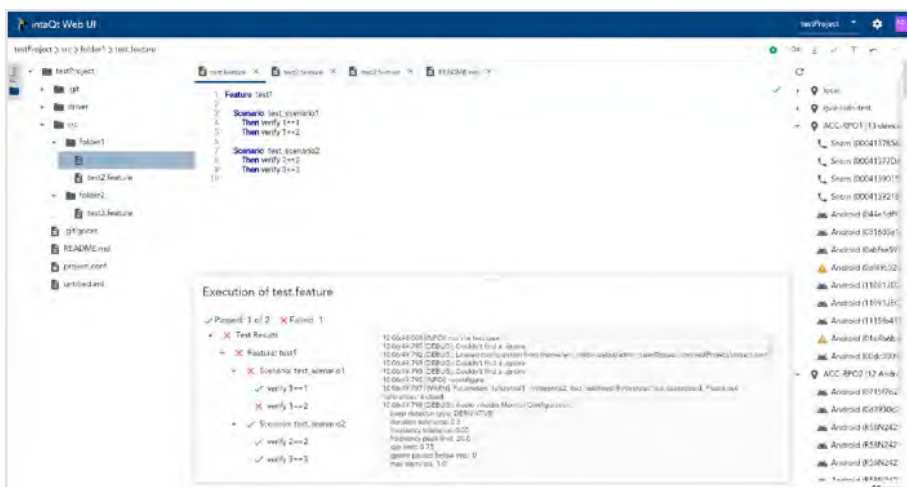
- Number matching - checks that two phone numbers match in a certain format
- Time matching - checks that a time occurs before or after a specified time
- Bit matching - checks that a bit's position corresponds to the bits of a given value converted to a binary
- Rounding - rounds values up or down in multiple ways including to full numbers, half numbers, to ceiling or to floor
- Date creation - create dates, with an optional date format specification



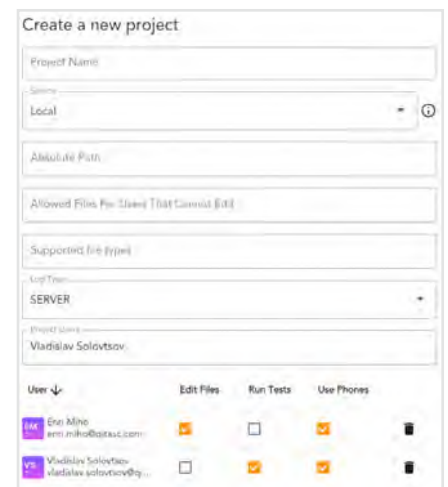
# intaQt web-ui

## Web-based User Interface

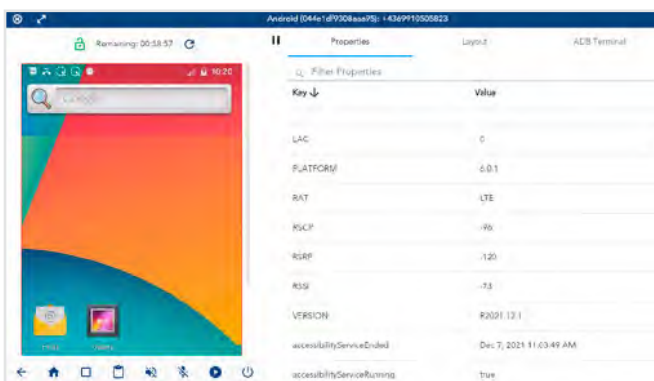
intaQt web-ui is a portal with a web-based user interface. It is a complementary product to intaQt studio, which would offer the option of web-hosted test services.



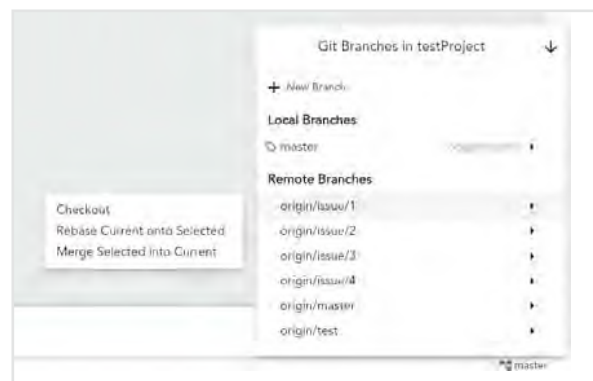
intaQt® web-ui main screen



user permission



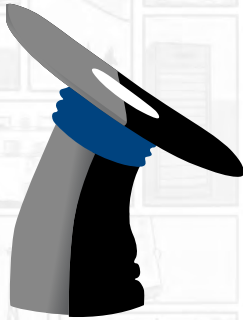
android phone



git

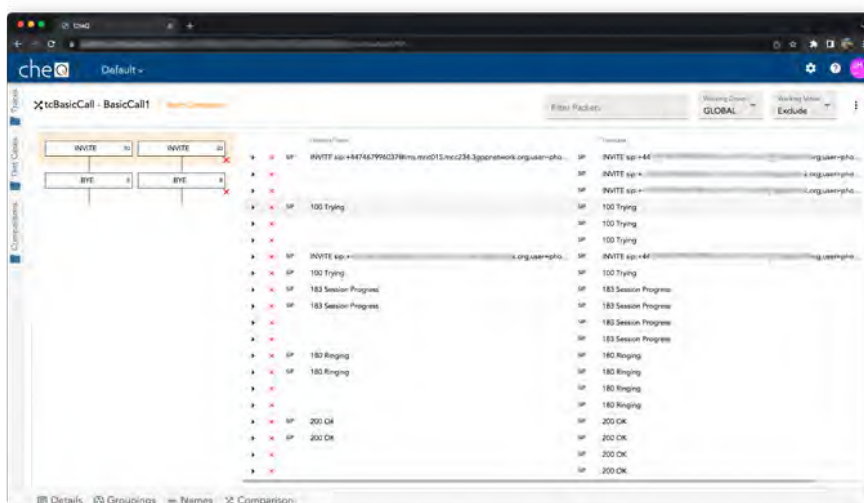
intaQt web-ui is a centrally accessible, secure web interface that provides an integrated development environment with many useful features to develop and manage test cases. The intaQt web-ui also

provides manual remote-control capabilities for android phones, VoIP phones and visualized POTS/ISDN clients. Git has been integrated to provide version management.

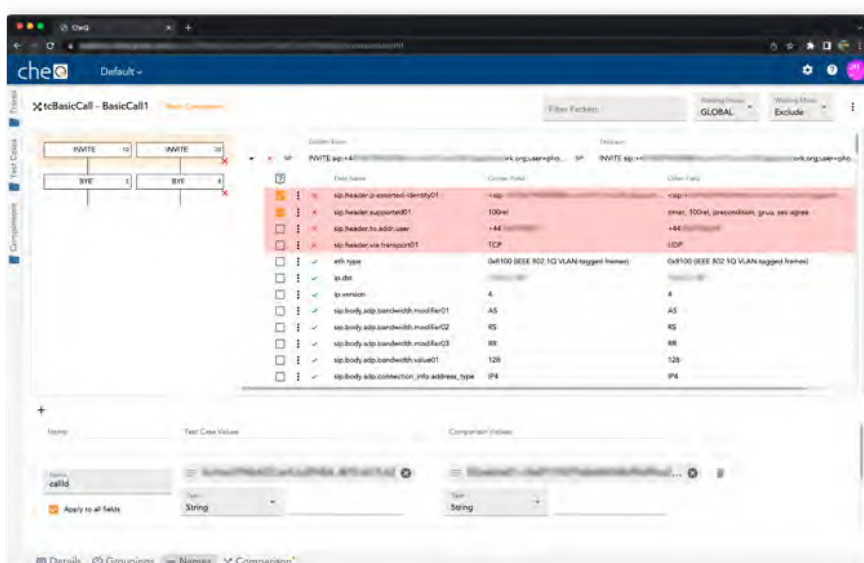


## Trace Comparison & Verification

cheQ is a trace compare tool for intelligent comparison of two traces. Trace verification can be customized by packaging of trace data and correlation rules, simply edited in the tool.



cheQ comparison between packets



Expanded cheQ comparison showing fields within a packet

Using cheQ, new traces are compared automatically. cheQ allows you to:

- customize trace verification by packaging trace data and correlation rules, simply edited in the tool
- easily copy and paste existing comparison rules to other test cases
- repeatedly compare this to new traces automatically
- keep a record of previous comparisons





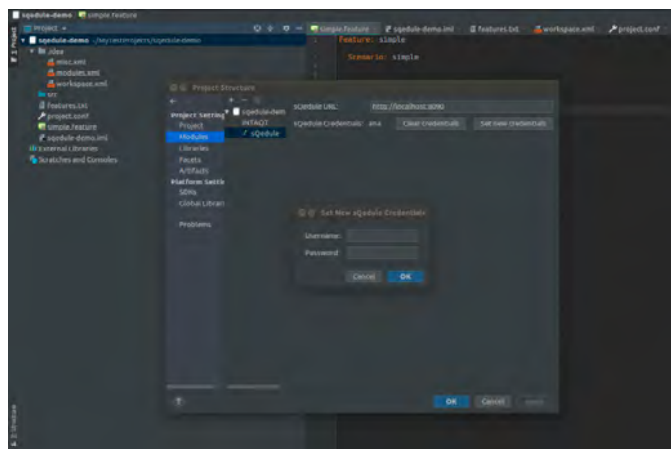
# sQedule

## Resource-aware, Intelligent Scheduling

QiTASC's sQedule is a resource-aware scheduler that schedules tests on intaQt agents. sQedule then evaluates and allocates phones to test cases based on availability and priority. sQedule always executes the maximum amount of possible test cases by optimizing the test cases' properties and available devices, which improves the volume of testing while preventing failures due to unavailable phones or incorrect test case selection.

## Cross-platform Functionality

sQedule is fully compatible with Windows, Linux and Mac OS. All the features and parameters listed below can be run and specified from within intaQt studio or via standard command-line interfaces.



## The sQedule UI

sQedule is available as a user interface within intaQt studio and as a command-line tool. The user interface lets users specify parameters so that tests are optimally executed. You can access by right-clicking on the project file, clicking on Open Module Settings or by tapping the F4 key. Click on Modules to type your username and password under Set new credentials.

The UI lets users define:

- Which tests (features) to execute
- When tests should be run

- How many times to retry tests upon failure
- What type of order (eager, random, or sequential) to use
- The maximum number of parallel test case executions
- Delays between test cases, such as for cool-down periods
- The sQedule server URL
- Optional authentication credentials

## Customizable Scalability

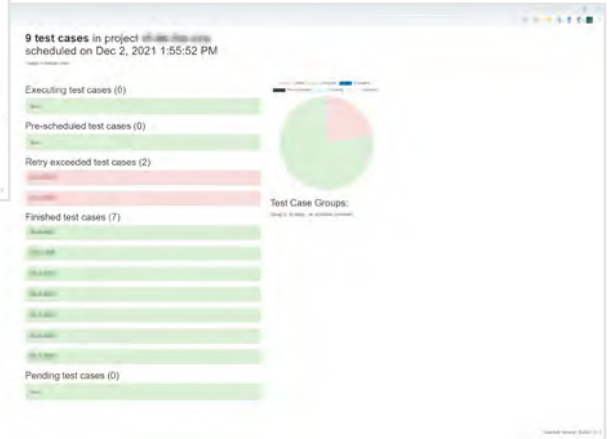
Resource-aware scheduling helps teams scale up their test execution volume. This is done by automating the selection, and prioritizing and scheduling tests that do not need human supervision such as regression tests. Furthermore, sQedule's resource-aware intelligence allows it to execute several magnitudes more of test cases than would be possible by a team of testers.

In addition to test case selection and execution, sQedule has been developed to:

- Assign the correct devices to each test case based on availability
- Ensure the optimal combination of test cases and devices is running at any given time
- Reduce or eliminate test case failures that occur when devices aren't available



the user interface of sQedule lets the user define which tests to execute, when they should be run, how many times, in which order, and so forth



## Reduce Testing Costs for your Team

Because sQedule takes over time-consuming manual activities such as test case assignment, scheduling, and prioritization, it reduces the number of intaQt® licenses that teams need. For small teams with limited budgets, this is especially helpful. For example, if you have a team of five testers but they aren't always running test cases at the same time, one or two floating intaQt® licenses and one sQedule subscription is a cost-effective alternative to purchasing individual licenses to cover each tester in the team.

Despite having fewer intaQt® licenses, your team will still see an improvement in its output. By allocating test suites to sQedule, it also leaves your testers with even more time to focus on complex tests and urgent tasks.

## The sQedule Client

The sQedule command-line tool provides the same functionality and customization as the UI. Teams that use continuous integration environments such as Jenkins or TeamCity can therefore incorporate sQedule into their builds to further increase levels of automation across projects.

Like the UI, sQedule CLI lets users define:

- Which tests (features) to execute
- When tests should be run

- How many times to retry tests upon failure
- What type of order (eager, random, or sequential) to use
- The maximum number of parallel test case executions
- Delays between test cases, such as for cool-down periods
- The sQedule server URL
- Optional authentication credentials

## Additional Features

sQedule also supports more advanced configuration options for more complex test case scheduling use cases, such as:

- Recognizing configuration switches or IoT devices
- Schedule constraints, such as time constraints where certain test cases are only run on even days or during the last quarter of an hour
- Test case grouping mechanisms, to automatically schedule constrained groups of test cases



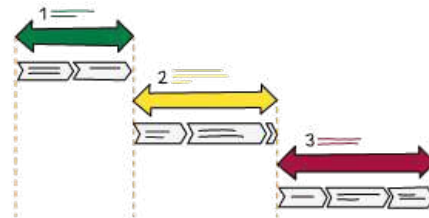
# conclude

## Integrated Reporting and Issue Tracking

conclude is a centrally accessible, secure web interface that collects, manages, and exports intaQt® test case project data and also includes automated bug and error tracking. conclude makes test data available to all users of a project including reports, logs, media attachments, and metadata. This eliminates the need for using spreadsheets, manually collecting test information, and reformatting it for recognition by additional databases.

## Platform and Browser Support

conclude is fully compatible with Windows, Linux and Mac OS. Currently it is supported by Mozilla Firefox and Google Chrome, with support for other browsers coming soon.



## Features for Test Managers and Project Managers

Test managers have access to all conclude features available to test engineers as well as:

- configure test categories (as seen on the right)
- approval workflow
- error category configuration

## Features for Administrators/IT

Administrators have access to all conclude features available to test engineers as well as:

- administer project settings
- external integration configurations

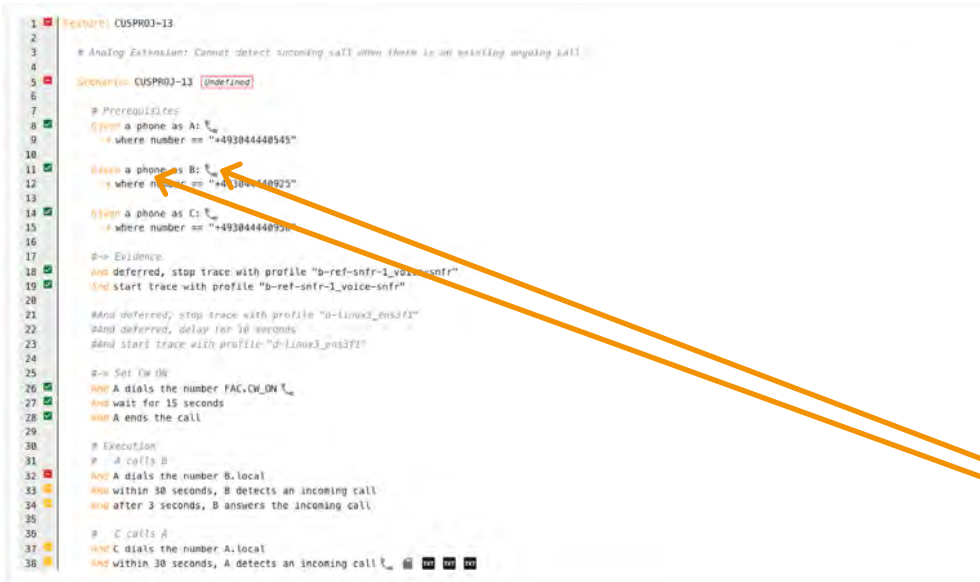


# intaQt

## Features for Testers and Project Members

Access test reports from your entire team in a centralized location, track bugs, and take a closer look at multimedia attachments, metadata, and logs.

- review & verify
- attachments
- bug and issue tracking
- charts
- data export














TEST RUN REPORT

TEST SUITE HISTORY

ERROR CATEGORY

UNASSIGNED REPORTS

Test Run Report

Test Name	Error Category	Passer	Start Time	Duration
 07.06.0070	Unknown	load09	16.02.2018, 04:13:55	4 minutes
 09.09.0519	Unknown	bet009	16.02.2018, 04:03:05	10 minutes
 09.01.0062	CDR verification Failed	load09	16.02.2018, 03:53:00	9 minutes
 02.09.0008		bet009	16.02.2018, 03:39:54	12 minutes
 02.09.0060		load09	16.02.2018, 03:28:31	11 minutes
 02.10.0101	Timeout Exception	bet009	16.02.2018, 03:20:15	8 minutes
 02.10.0112		load09	16.02.2018, 03:16:00	2 minutes
 02.10.0121		bet009	16.02.2018, 03:09:44	6 minutes
 02.07.0004	Timeout Exception	load115	16.02.2018, 03:09:45	4 minutes
 02.10.0144		bet009	16.02.2018, 03:03:57	5 minutes
 02.10.0137	Unknown	load006	16.02.2018, 03:02:14	0 minutes



**Charts:** View visualizations of test data to get a quick overview of your project's progress. Charts provide details of the daily test run history, pass and fail rates, and defect incidence.

**Review & Verify:** The Test Run Reports feature lets users view and review a project's test reports and their metadata. Because conQlude collects test results and artifacts from all members of a team, users can access reports and data from the tests they executed as well as those of their teammates.



**Attachments:** Download or preview test attachments, including media, text, XML, logs, and step-specific metadata. conQlude even lets users listen to audio recordings, view screenshots, and access metadata in its original format.

TEST CATEGORIES	HPQC TEST SETS	HPQC SETTINGS
<b>Test Categories</b>		
Name	Tag	
Telephony	@phone	
Webtest	@web	
Apptest	@app	
+		

**Configure Test Categories:** conQlude automatically recognizes test categories using tags from intaQt® feature files. Categories can also be changed or added within conQlude. These configurations can also be mapped to projects and configurations in external issue track and test management systems.





# mimiQ

## Simulator for Network Elements

mimiQ lets you define https message sequences. It can act as a simulator for a network element in telecommunications networks or provisioning systems. It is also operated as a load and stress test system, especially for 5G Nchf, Npcf, etc. interfaces.

```
charging_server.rhai
6
7 let request = env.serve_http();
8
9 assert(request != ());
10
11 print("Request uri: " + request.uri);
12
13 reqBody = request.get_json_body();
14
15 if reqBody != {} && reqBody.value("notifUri") != {} {
16   PCF_notifUri = reqBody.value("notifUri");
17 }
18
19 if "/update" in request.uri {
20   sendRequest(env, request, PCF_notifUri);
21 }
22
23 if "/nchf-convergedcharging/v2/chargingdata" in request.uri {
24   sendResponseV40(env, request);
25 }
26
27 if "/nchf-spendinglimitcontrol/v1/subscriptions" in request.uri {
28   sendResponseV28(env, request);
29 }
30
31 }
32
33 fn sendResponseV28(env, request){
34   let headers = #{}
35   "Content-Type": "application/json",
36   "Location": "SpendingLimitControl-12345"
37 };
38 let args = #{};
39 let responseCode = 201;
40 let body = "";
41 if request.method == "POST" {
42   responseCode = 201;
43   let subID = env.get_str("subscriberId");
44   args = #{}
45   "subscriberId": subID,
46 };
47 body = env.template("mimiQ/templates/http/dataCharging/spending_limit_control_create.json", args);
48 if request.method == "DELETE" {
49   responseCode = 204;
50   args = #{};
51   body = "{}";
52 };
53 request.respond_with(responseCode, headers, body);
54 }
55
56 fn sendResponseV40(env, request){
57   let headers = #{}
58   "Content-Type": "application/json"
```

Example mimiQ 5G data changing scenario script

mimiQ is a more general-purpose "interface testing tool". It is a test script executor with built-in support for any kind of http communication. mimiQ supports both test scripts and load testing. The initial focus is on HTTP, especially for 5G.

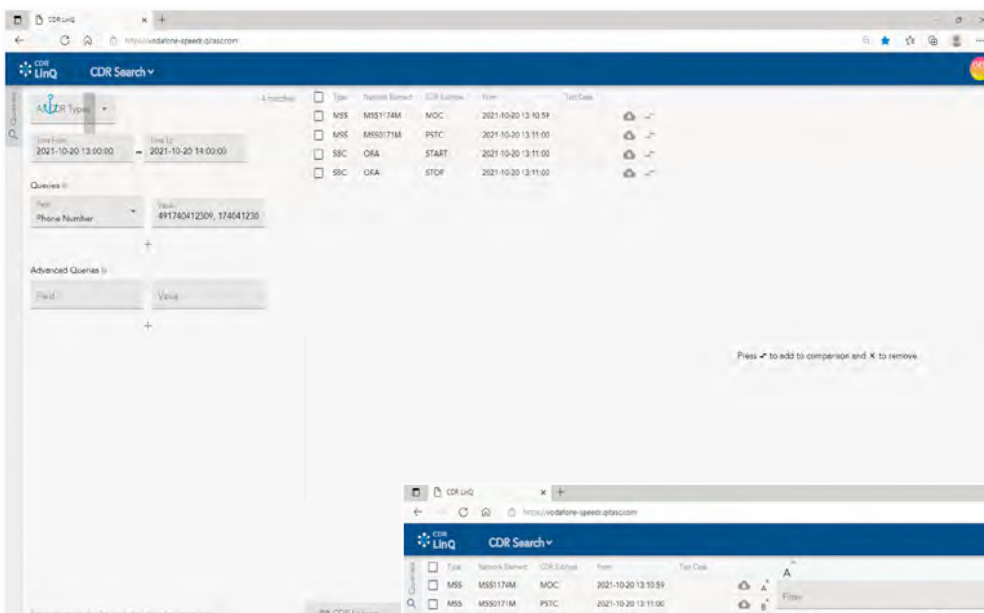
mimiQ is extremely fast and has targeted features for load testing and a scripting language. It provides a templating engine so that scripts can easily load request/response data. With a very fast execution engine we can generate load and stress test scenarios.



# cdr-lin

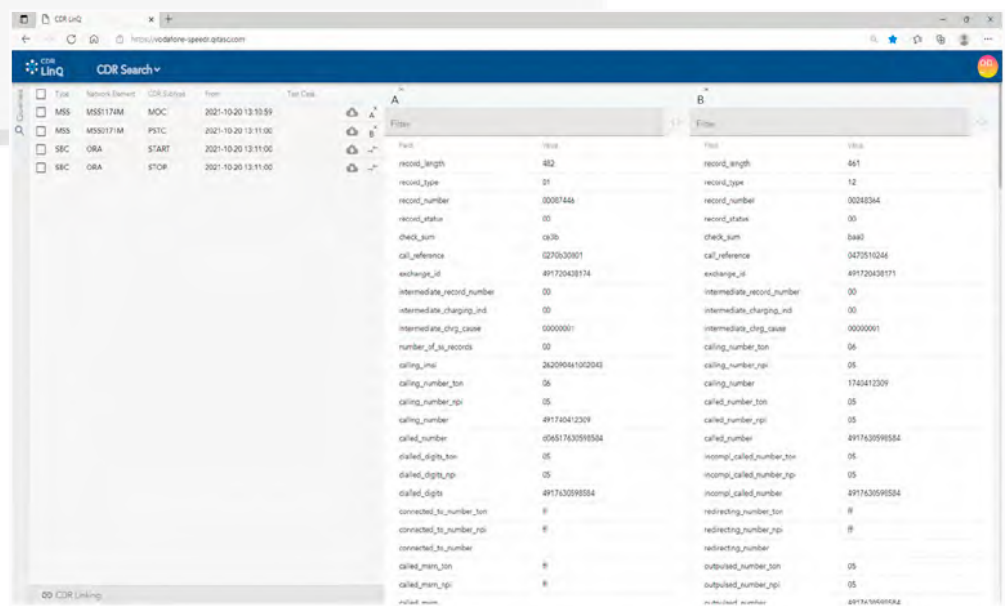
## CDR Web Service

Cdr-lin is a web service for collecting, filtering and presenting CDRs which have been generated by test cases. CDRs written by network elements are collected and decoded. Cdr-lin shows two CDRs and compares them.



Left window: Based on the criteria defined on the left side, CDRs found and listed on the right

Right window: CDRs can be viewed online by clicking on it, maximum 2 CDRs next to each other



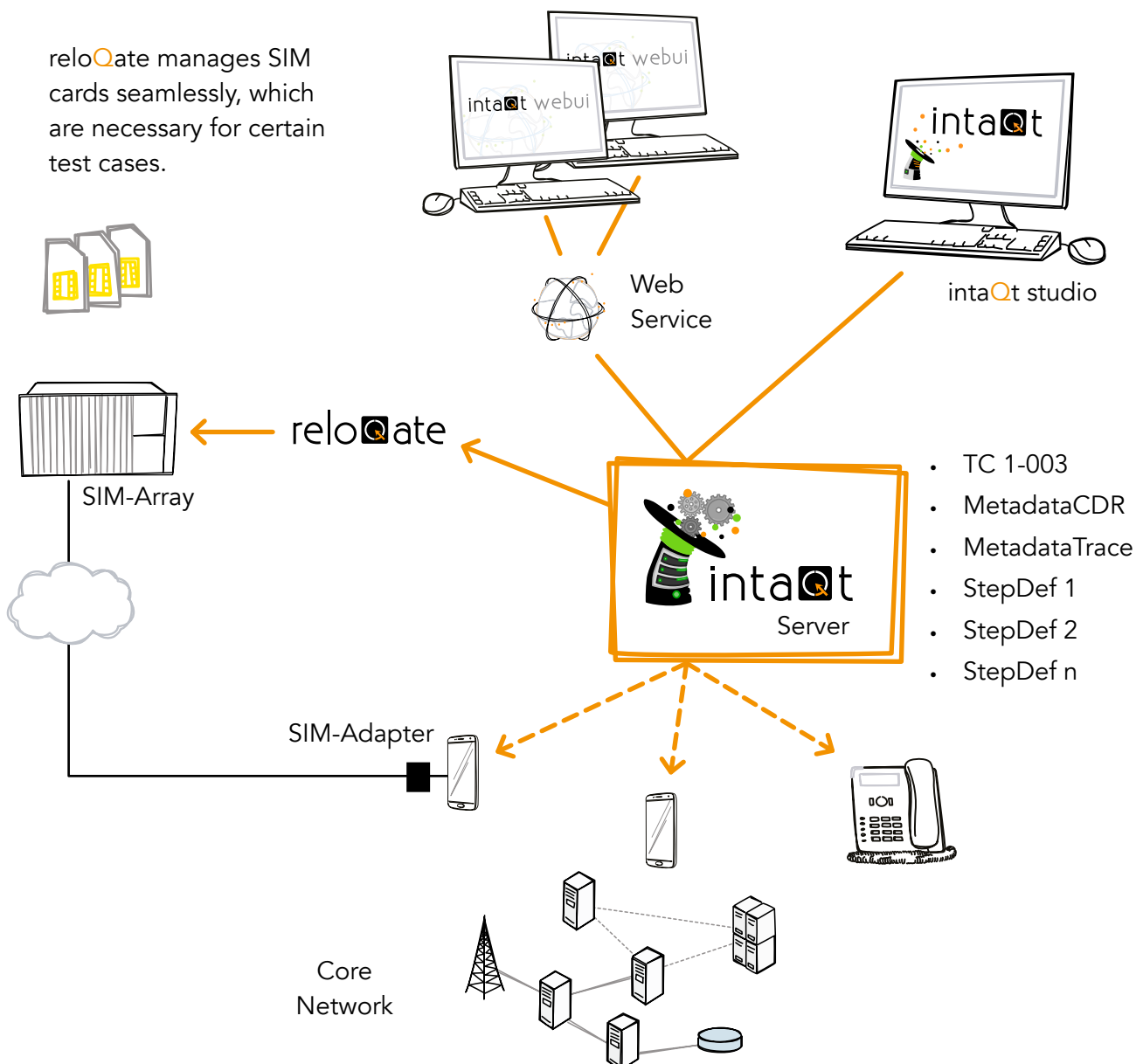


# relocate

## SIM Card Detector

relocate is a special background service for mapping SIM cards located in a SIM array/SIM Mux to a dedicated phone. The test expert does not need to do anything. relocate detects if the required SIM is already mapped to a phone or not, maps the SIM to a phone, reboots the phone, and makes it available for the test case.

relocate manages SIM cards seamlessly, which are necessary for certain test cases.

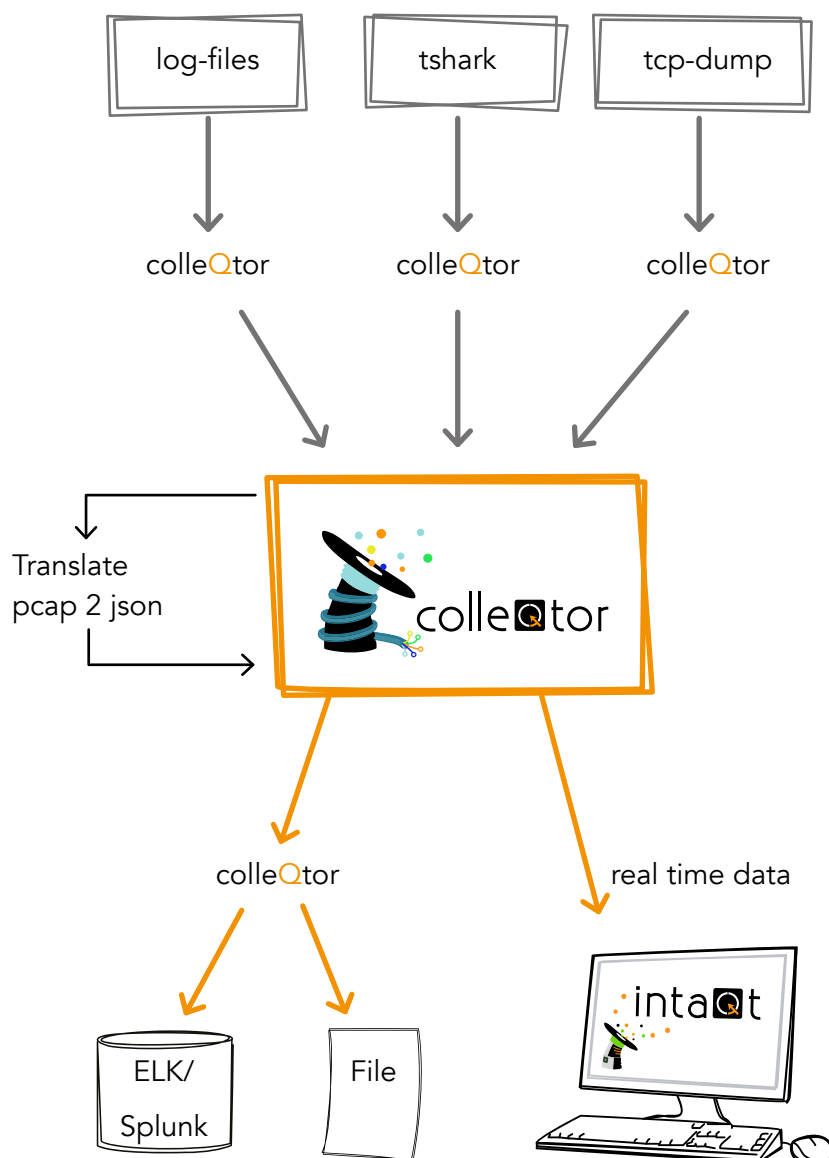




# colleQtor

## Evidence Collector

The colleQtor is a service for collecting evidence which comes as a stream and is then provided to the test case for verification purposes.



colleQtor allows you to:

- collect various trace data in real time
- intaQt® integration to be used for test case verification/processing in real time, while the test case is running
- Export of data into file/ ELK/etc.

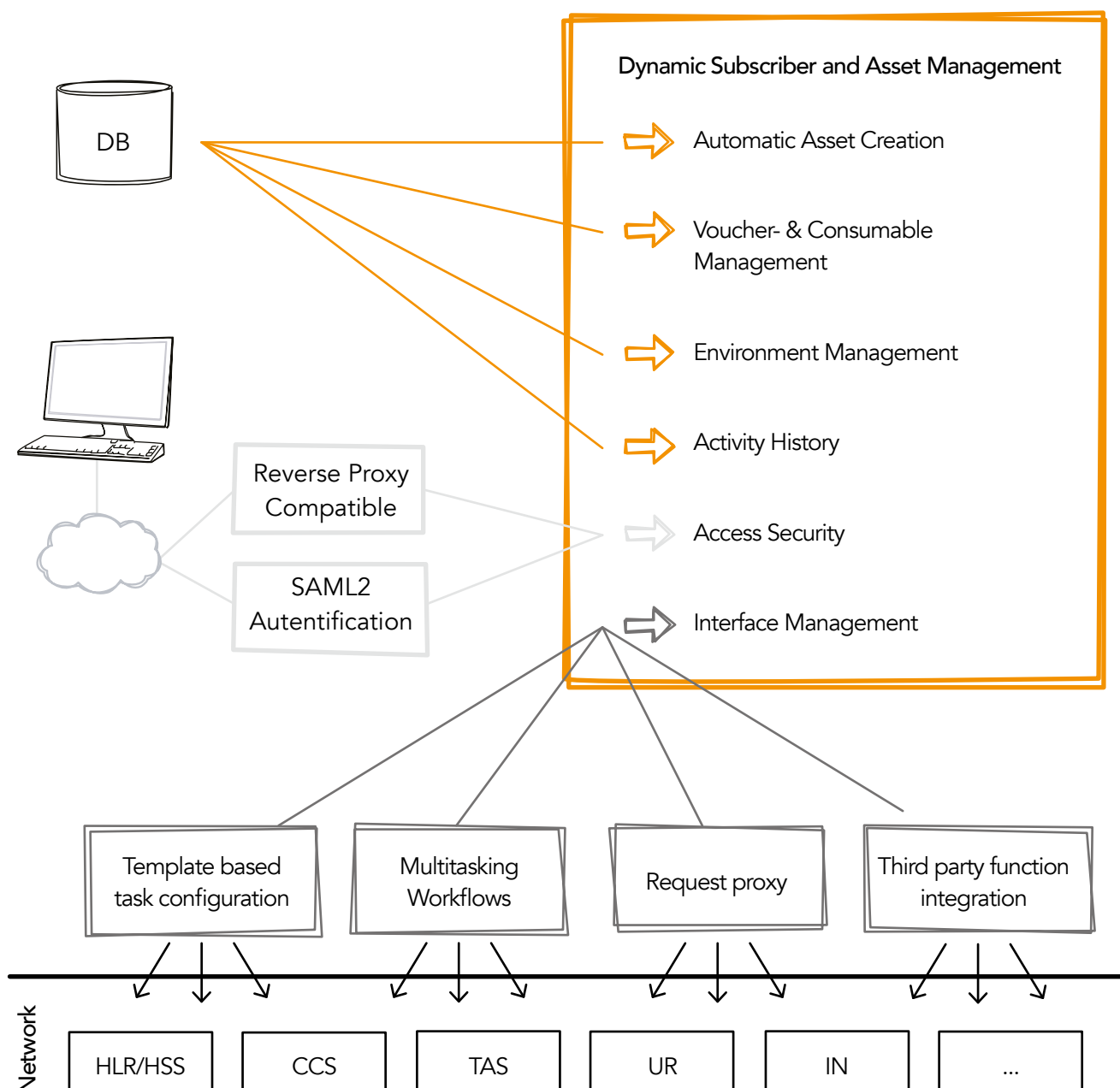




# subsQriber-db

## Data Store

subsQriber-db is a data store for flexible SIM and device asset management, profile management and assignment, voucher management, multi-environment support such as transparent request proxy, request templates, multi sequence templates, and GUI-driven request execution, execution, voucher and request history, permissions at subscriber level, SAML2 integration.



"Quality is never an accident.  
It is always the result of intelligent effort."

(John Ruskin)



Get in touch!

Contact us

+43 1 810 21 73  
info@qitasc.com

Visit us at

Diefenbachgasse 53  
1150 Vienna / Austria