

HIP Release 1.0

This document gives a brief overview about the scope of **Hosting Integration Platform Release 1.0** (release date: 07.08.2012).

The primary target of this release is to make important HIP Features available on our **DEVELOPMENT** Integration Environment so that other Hosting product teams (like Domains SSA) can start development / integration with HIP Features.

Table of contents:

What is the 1&1 Hosting Integration Platform?	2
CloudIA.....	2
Development Integration Environment (DIE)	2
CloudIA SDK.....	2
Documentation	3
Guidelines.....	3
Authentication Facade (HIP Login).....	3
Data Storage.....	3
BSS Contract Management	3
HIP Registry	4
OAuth2 Authorization Provider	4
RAIN.....	4
Asynchronous rendering engine.....	4
Messaging support.....	5
Modular RAIN SDK	5
I18N	5
CSS Media Queries support	5
Visual elements.....	5
Security.....	6
Exception handling.....	6
HTTP Session	6
1&1 Developer network.....	6
Governance	6

What is the 1&1 Hosting Integration Platform?

The 1&1 Hosting Integration Platform ([HIP](#)) is a multi channel platform designed to simply and efficiently build and integrate highly scalable, robust and extensible enterprise services, systems and applications into the distributed, heterogeneous 1&1 Hosting system landscape.

[HIP](#) comes with [RAIN](#) (Rapid Application Integration) a frontend platform designed to build enterprise web applications that can run on desktop and mobile devices and [CloudIA](#) (Cloud Integrated Architecture) a backend platform that provides cross cutting functionalities for developers to build, distributed and composite systems, services and applications within a consistent [RESTful](#) service oriented environment.

CloudIA

CloudIA (Cloud Integrated Architecture) addresses the backend integration of a Product Service System (PSS) and offers cross-cutting functionalities. It is the backend layer of HIP. For a more detailed explanation see <http://developers.1and1.com/hosting/hip/pages/overview/what-is-hip.html#cloudia>

Development Integration Environment (DIE)

The DEVELOPMENT Integration Environment is accessible for other PSS teams to do early integration tests of their system with our HIP services or test PSS-to-PSS communication scenarios. The infrastructure setup will be similar on follow-up stages (AC1, PROD). Development however is recommended to be done with our CloudIA SDK which allows needed HIP services to run in a local sandbox environment (no network dependencies/issues).

- Stable Dev environment (VMs, DB-cluster, ...)
- GSC as development base
- Continuous Integration (incl. release process)
- Automated integration tests (Fitness: <http://vm1786.development.lan:9000/>)
- System monitoring as required by IT-Ops
- IT-Security checklist for our applications
- RESTful API Guidelines applied on all services

CloudIA SDK

The CloudIA SDK allows local development of your PSS backend services. Therefore all needed HIP services run in a local sandbox environment so that there are no network dependencies/issues. We also offer Maven archetypes to setup a new PSS as well as a sample app where you can see how things are used / tied together. We will show this in our CloudIA Coding Event.

- PSS sample app with best practice solution
- Demonstration of SDK in CloudIA Coding Event (on 09.08.2012)

- Maven archetypes for simple PSS setup / generation
- OAuth2 authorization model support
- HowTo / tutorials for HIP integration
- Ready to use for local development

Documentation

Here you can read all about our Hosting Integration Platform:

- <http://developers.1and1.com/hosting/hip/index.html>
- Infrastructure documented
- HIP services documented (REST API, system context diagram, UML)
- Operations manual for all services (aligned with Hosting-OPs)

Guidelines

We provide a Guidelines section in our documentation which shall leverage development of „state-of-the-art“ PSS following best practices at 1&1 Hosting Development. Some guidelines are contributed by our team (like the Logging Guideline).

- REST API Guidelines (thoroughly used, not provided)
- Logging Guideline
- I18N Guideline
- Caching Guideline
- OAuth2 Authorization Guidelines

Authentication Facade (HIP Login)

As PSS developer you will most likely use this login functionality via RAIN frontend. The long term goal is to provide “Authentication as a Service” for all 1&1 systems. Currently we provide basic support for 1&1 Account. Login with customer number is coming soon.

- Login functionality using UAS (IDP, NS), including facade of user management
- Mapping 1&1 Account to Customer (currently mocking services from Munich)
- Error handling for Login and Logout

Data Storage

The current stage of expansion of our Data Storage is used as Session Store in all RAIN based Self Service Apps (SSA). It is (of course) possible to access this storage via REST interface directly. Further development will scale up this storage to a internal “Storage as a Service” offering.

- Session Store (CRUD, Bulk operation, data security via OAuth2)
- Data Storage Criteria API with support for filtering and paging of data
- Data Storage Cluster using a highly scalable NoSQL database (Cassandra)

BSS Contract Management

The Business Support System (BSS) Contract is our first citizen to deliver contract information via new RESTful interfaces, see <http://developers.1and1.com/hosting/hip/pages/overview/who-uses-hip.html>

- Contract selection with RAIN frontend integration
- Provisioning Information Manager supports
 - caching, filtering, paging of data
 - articleItems from migraene for legacy migration
 - provisioningItems for new SSA

HIP Registry

The HIP Registry is the central place for all PSS teams to resolve their service dependencies. There you can lookup other PSS endpoints or register your own system(s). It brings more flexibility into the 1&1 Hosting Environment since deployment of a system is independent from other PSS (no more hard-wired service dependencies).

- Service registry used for HIP internal services (CRUD, filter, ...)
- Register PSS Services
- Request Broker aka service lookup functionality (with advanced filtering)

OAuth2 Authorization Provider

At a first glance this might seem to „complicate things“ for you. Why? Almost none of our services can be accessed without a proper OAuth2 access token. We have extensive documentation and code samples/archetypes that help PSS teams to apply OAuth2 security on their systems (so they can open up to 3rd party). We also offer a HIP Authorization Service where you can register your protected resources. Further development will deliver simple to use but still high secure “Authorization as a Service” for you.

- Documentation (How To, Use Cases mapped to OAuth flows)
- PSS Archetype with OAuth2 support (see SDK)
- All HIP Services secured (including fine grained client config)

RAIN

RAIN (Rapid Application Integration) addresses all topics around PSS frontends (often called “Apps”). It is the frontend layer of HIP, see

<http://developers.1and1.com/hosting/hip/pages/overview/what-is-hip.html#rain>

Asynchronous rendering engine.

Tired of waiting 10 seconds for a web app to show its content? The asynchronous rendering engine of RAIN allows to speed it up tremendously.

- Asynchronous rendering: shows (partial) content when its available
- Scale-up: RAIN dynamically decides where to render content (server side or client browser). This can for example vary for the device used (PC versus Smartphone).
- Scale-out: just add more servers for server side rendering
- Server side API, Client side API

Messaging support

Allows “classical” event-based communication but also to the client side (push)!

Even more innovative is the intents mechanism for web apps (ok, Android had them before:

<http://www.vogella.com/articles/AndroidIntent/article.html#intents>)

- Client side publish / subscriber
- Intents mechanism (client / server)

Modular RAIN SDK

The RAIN SDK is your starting point to develop a new SSA (Self Service App). A new project directory is automatically filled with all configuration files, default content (html page) etc. so that your new “component” works out of the box and you can view it in your browser

- Create project
- Create component
- Start / Stop server
- Generate localization files

I18N

RAIN supports Internationalization / Localization of web apps out-of-the-box. It provides tooling to generate localizable content (e.g. PO files) ready for translation by our Technical Editors (process is already approved by them). Rain offers:

- Text localization
- Images localization
- Layout localization
- Dynamic language selector
- Server side / client side support

CSS Media Queries support

RAIN support new CSS 3 features like Media Queries (but also fallback mechanisms for older browsers).

Visual elements

Currently few content assets are shared between our 1&1 products. This means every product has its own solutions for visual elements like list boxes, tables etc. RAIN will provide a 1&1 component repository from start on. So every PSS will benefit from a standard set of elements to assemble its Frontend UI. In Release 1.0 we already have following elements (more to come). You can have a look on RAIN Demo page: http://vm2384.development.lan:1337/sprint_example_list/index

- Button
- Drop down button
- Drop down menu
- Checkbox
- Radio button

- Group of checkboxes
- Group of radio buttons
- Progress bar (quota indicator)
- Datagrid with paging / sorting / filtering support
- Text field
- List box component
- Contract selector

Security

RAIN covers the authentication and authorization aspects for all web apps. It uses the corresponding parts of CloudIA (see HIP Login, OAuth2 Authorization).

- 1&1 Account login component
- OAuth2 integration with CloudIA
- Authorization for frontend components (compliant with Role Based Access Control v2.0 standard)
- Dynamic security conditions (pluggable security hooks specific to each application).

Exception handling

Class library with generic error handler (including logic for communication with RESTful backends).

- Custom exception pages.
- Error handling generic handler.

HTTP Session

RAIN offers session handling (one shared session for all SSA). It is secure to use since all apps can read session data but only write in their own “namespace” (using valid OAuth2 tokens). In later releases we will offer connection to “externally” hosted sessions through Single Sign On (SSO).

- Pluggable storage with default CloudIA data store integration (OAuth2 secured)
- Open source storage for http session (Memory store)

1&1 Developer network

RAIN also offers extensive documentation about its features: <http://rainjs.github.com/rainjs/>

- Documentation and how to articles
- API Documentation
- Sample applications

Governance

This section helps frontend developers with documentation of SSA development best practices (counter-part of the CloudIA guidelines for backend developers).

- Nodejs programming best practices
- Logging guidelines