Draft for a user centered digital rights management system[1]

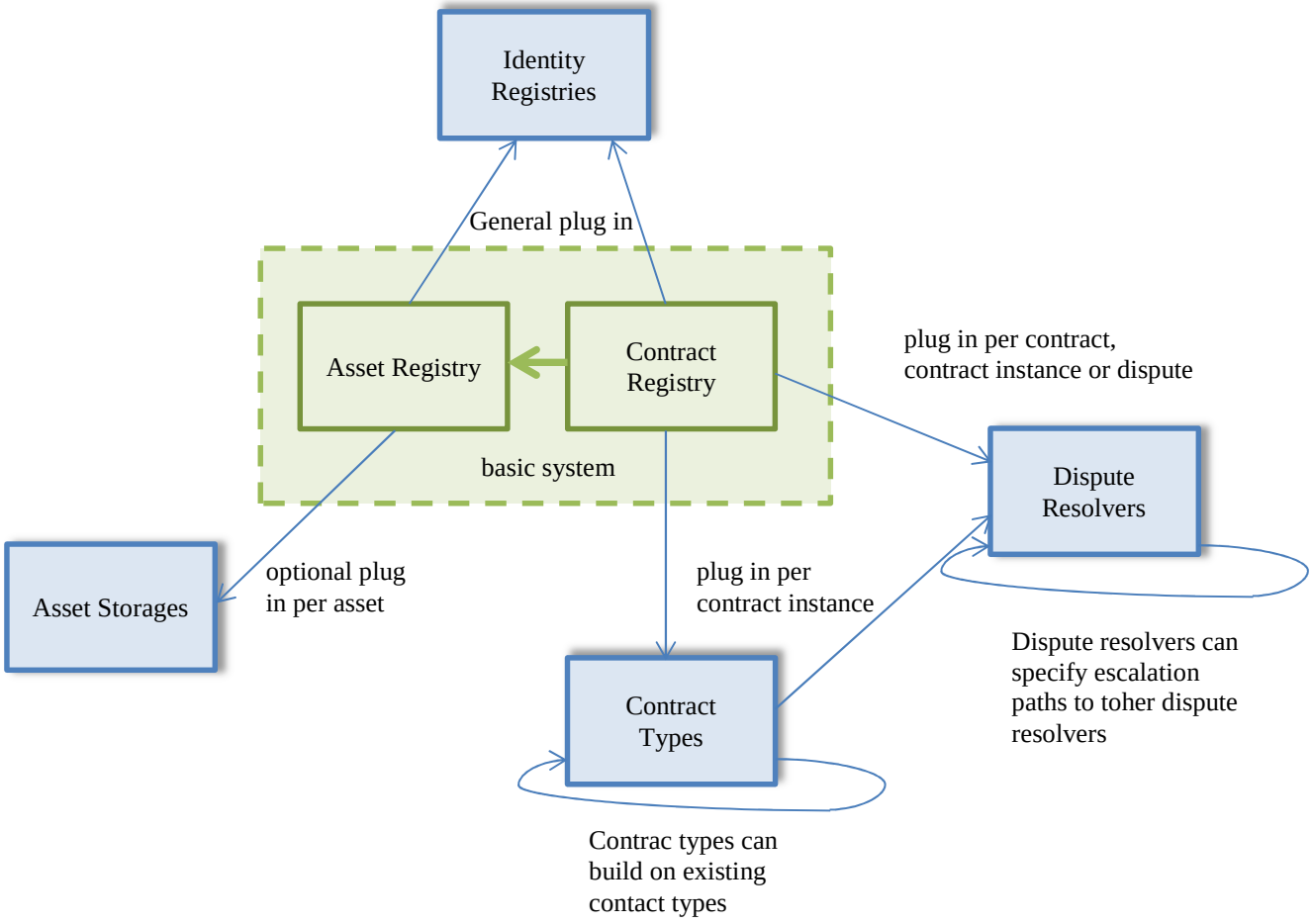
Version 0.1 December 4th 2014, Jörn Erbguth (joern@erbguth.net)

Basic Architecture

The system has a modular architecture consisting out of

- an asset registry - this should only exist once to detect duplicate registration
- a contract regsitry- this also exists only once
- contract types - there shall be many
- identity registries - there shall be many as well
- dispute resolvers - there shall be many as well

The file registry and the basic contract layers are provided as the backbone of the system. Different contracts can be plugged in on a per asset base. A contract specifies not only the terms of use but also the revocability, transferability etc. of a license. Identity registries provide a means of assigning a real person or legal entity to a blockchain id in a more or less certified way. A certified identity will be the requirement for certain functions (e.g. dispute a right claimed by somebody else). Dispute resolvers can be assigned to a contract, to a contract instance or to a dispute of a contract instance.



[1] Based on the Hackathon organzized by Jean-Henry Morin in Geneva on November 26th 2014

Asset Registry

Every asset needs to be registered. The registry will be done with a hash code of the file. Everybody can register a file. The person who registers will add a motivation for the role why they register a file like beeing the creator of it or refering to it because they created derivate works based on it. For derivative works the source assets are referenced.
With the registration you can proove that you had access to the file at a given time (timestamp). If you choose to identify yourself, everybody can see that it was you who registered that file.
An asset storage can be linked to the registry if you want the file to be accessible.
The asset registry provides the asset entities for the contract registry.

Contract Registry

With a contract registry person/legal entity A claims that they own a certain right and can transfer that to person/legal entity B. Every contract has a starting date. Contracts can have an ending date. The scope of the rights, obligations of both sides, the terms of the contract (revocability, transferability, modifyabilities) are specified by referencing a contract type. Contract types can store key-value-pairs in a contract instance to store specific parameters. Contract types provide an interface so that the contract registry can query whether transfer and revocation is possible and who has to consent to it.
Contract types can reference other contract types in order to inherit properties from them.

Dispute Resolvers

There might be disputes over the ownership of an asset. There may also be disputes about contracts. A dispute resolution entity can be referenced for a contract type, individual contract instances or with individual disputes. Dispute resolvers provide a certain automated workflow for dispute resolutions. They have the power to overturn contract type rules. Dispute resolvers can also reference some arbitration entities or jurisdictions to settle a dispute. Dispute resolvers can refer to other dispute resolvers in order to escalate or settle a dispute.