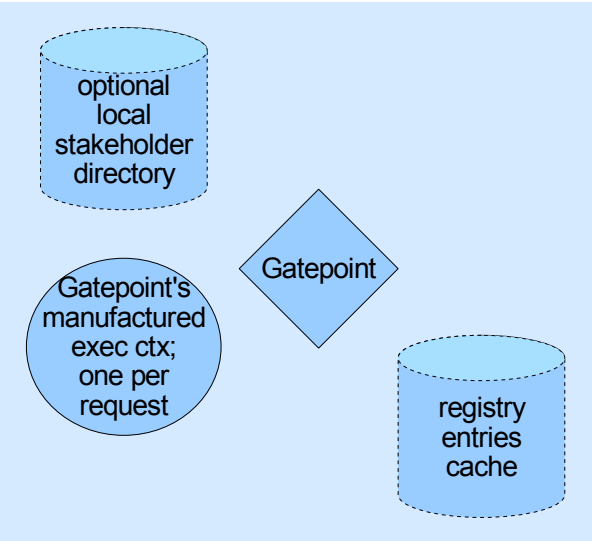
Components in a deployed CDCL Gatepoint installation;
dotted lines indicate optional or transient items;
the exec ctx is present only at a run-time redaction event

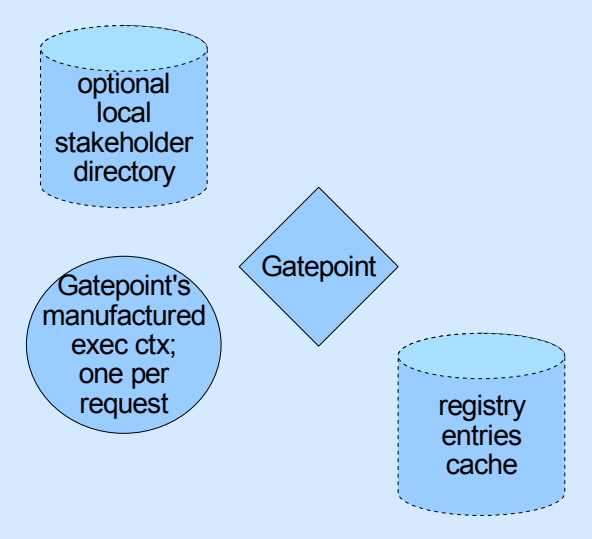


The receiver is a party other than the client who will get the resulting redacted document

(future) receiver of "forwarded" redacted document

A third party PEP is feasible, but it would be invoked not from a Gatepoint but rather from some entity with a document containing decision mark-ups, such as a Gatepoint *client*

optional 3rd party policy enforcement point

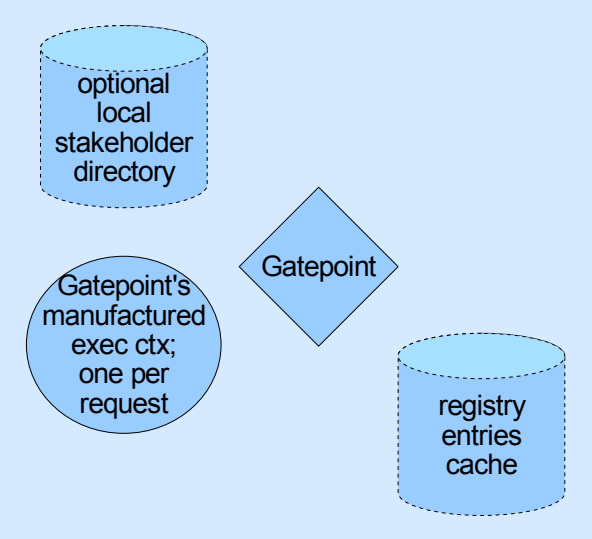


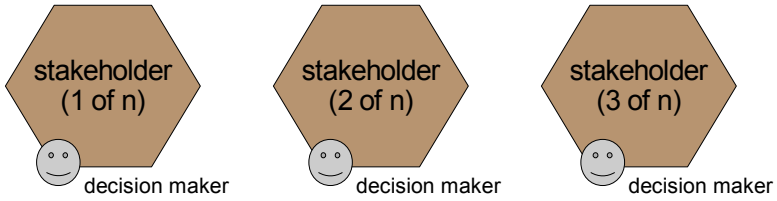
client requesting redaction of document

(future) receiver of "forwarded" redacted document

optional 3rd party policy enforcement point

For now, the client will be the entity that receives the resulting redacted document and/or any response meta-data for the request transaction



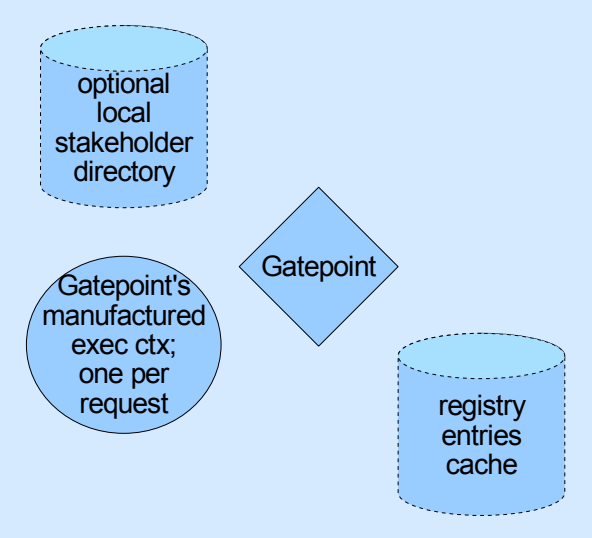


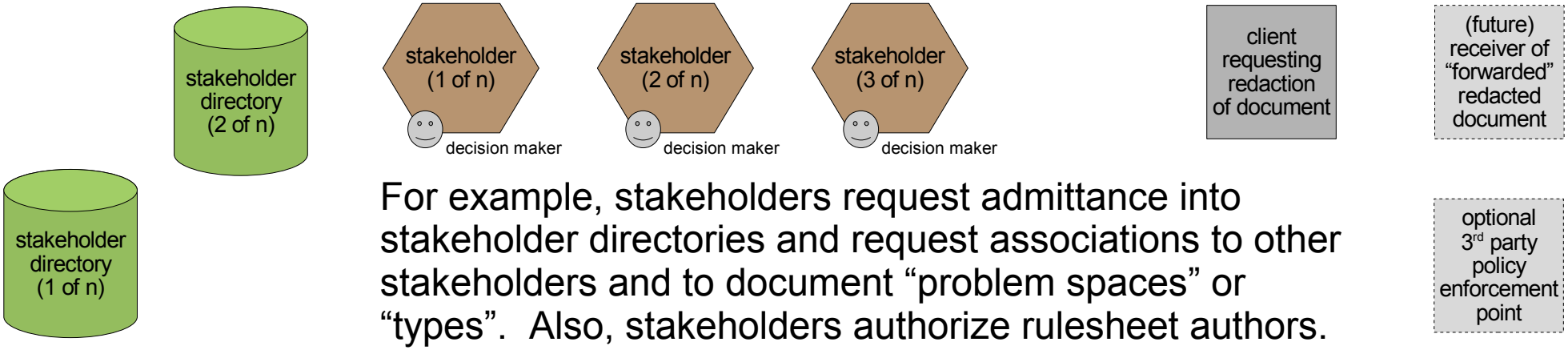
client requesting redaction of document

(future) receiver of "forwarded" redacted document

Stakeholders don't have any direct involvement with a Gatepoint, but they are the ultimate decision making authorities, along with various server administrators, for declaring most dependencies within a redaction "problem"

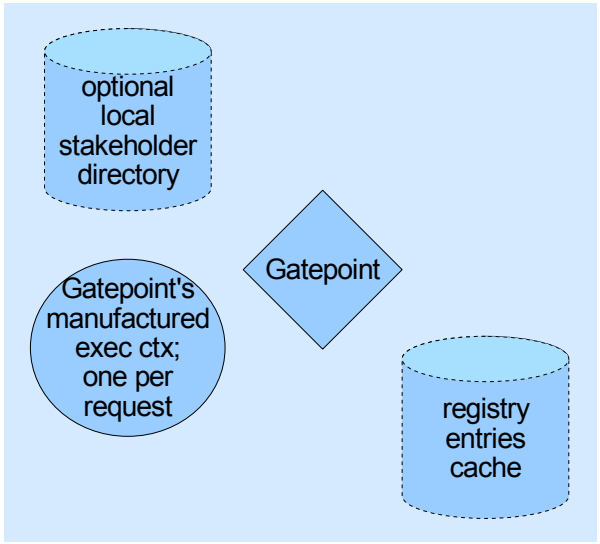
optional 3rd party policy enforcement point

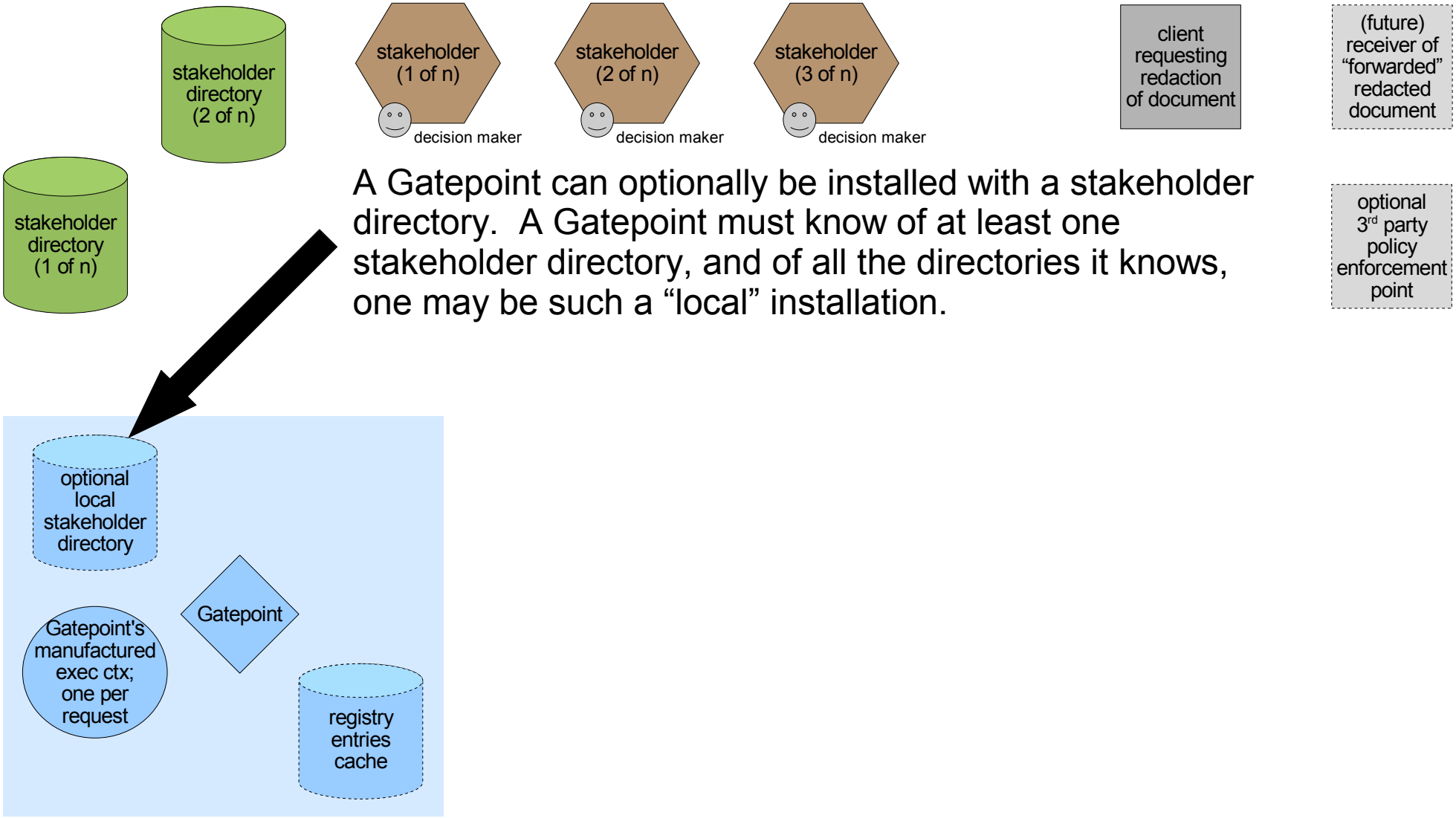


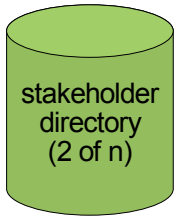


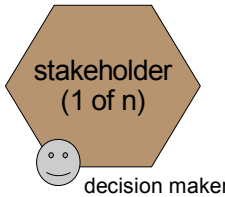
For example, stakeholders request admittance into stakeholder directories and request associations to other stakeholders and to document "problem spaces" or "types". Also, stakeholders authorize rulesheet authors.

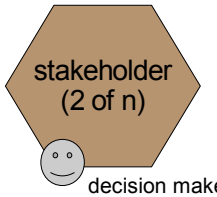
This information is all held within stakeholder directories

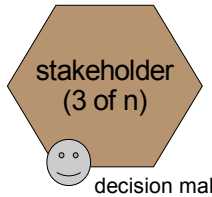


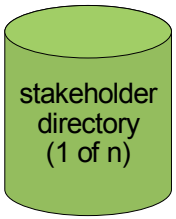

stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
A Gatepoint can optionally be installed with a stakeholder directory. A Gatepoint must know of at least one stakeholder directory, and of all the directories it knows, one may be such a "local" installation.
optional 3rd party policy enforcement point
optional local stakeholder directory
Gatepoint
Gatepoint's manufactured exec ctx; one per request
registry entries cache



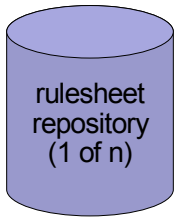


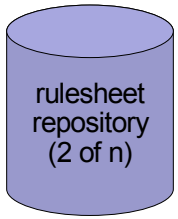


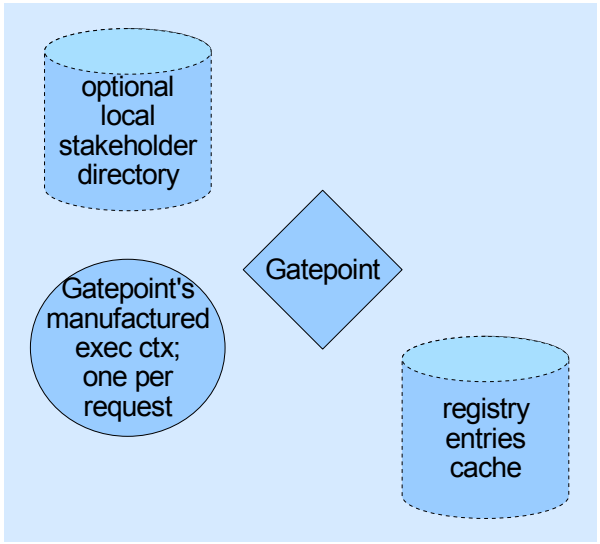


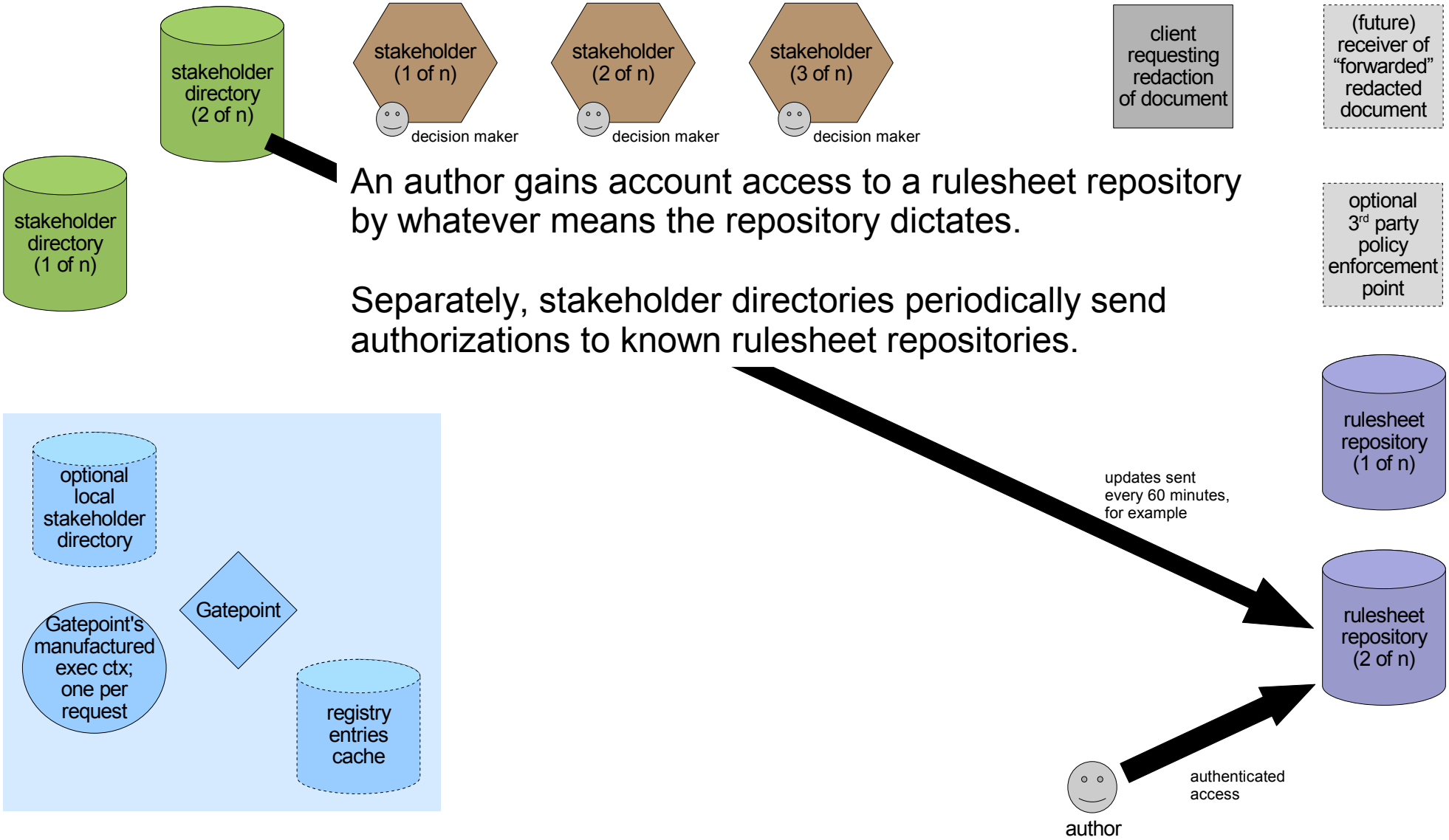


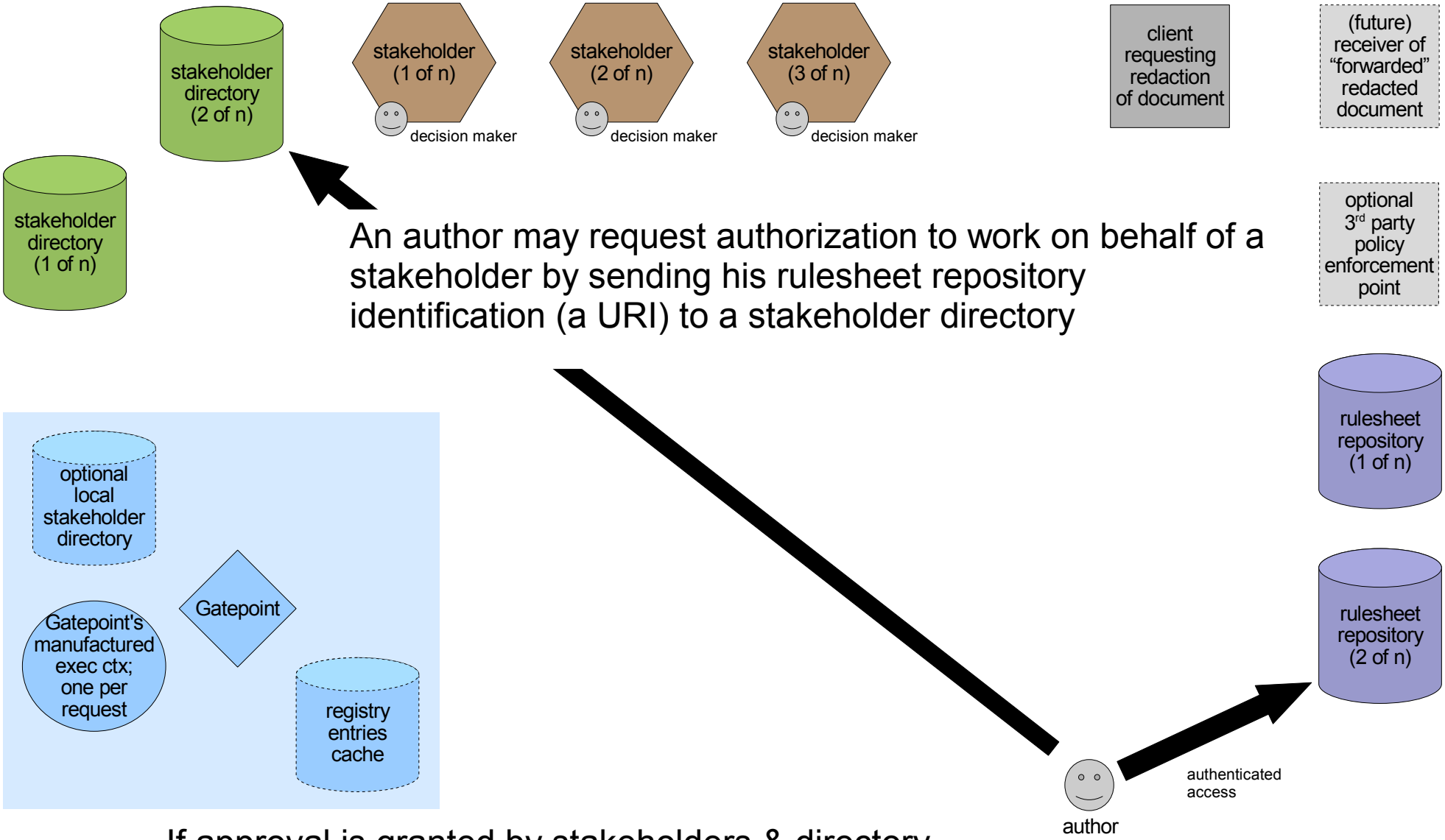
Rulesheet repositories are version control systems for CDCL rulesheets written by authors. Through stakeholder authorizations held in stakeholder directories, authors may declare associations between rulesheets and stakeholders, and these associations are stored within the rulesheet repositories as rulesheet meta-data





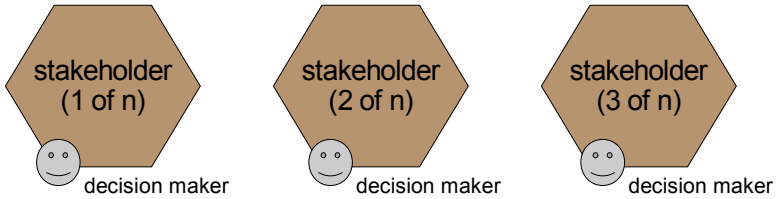



stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
An author gains account access to a rulesheet repository by whatever means the repository dictates.
Separately, stakeholder directories periodically send authorizations to known rulesheet repositories.
optional 3rd party policy enforcement point
optional local stakeholder directory
Gatepoint
Gatepoint's manufactured exec ctx; one per request
registry entries cache
rulesheet repository (1 of n)
updates sent every 60 minutes, for example
rulesheet repository (2 of n)
author
authenticated access



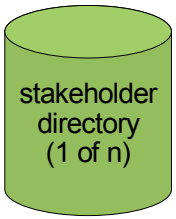
If approval is granted by stakeholders & directory administrators, the authorization for *this author* at *this repository* for the *specific stakeholder* is persisted in the directory and then included in subsequent updates the directory sends periodically to the repository.

The author may thereafter associate rulesheets to that stakeholder.



client requesting redaction of document

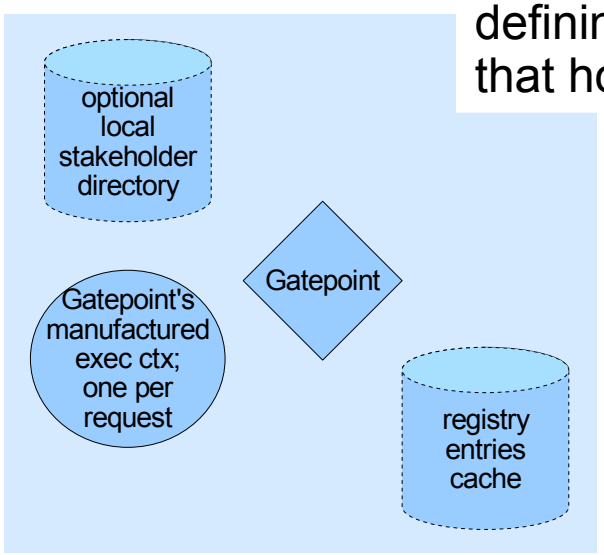
(future) receiver of "forwarded" redacted document

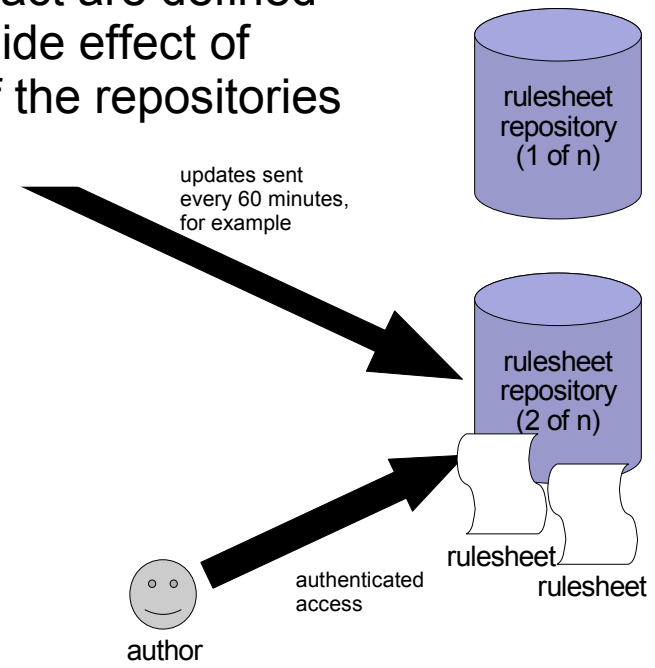


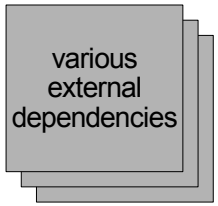
Based on the authorizations arriving from directories, the author creates rulesheets and associates them to stakeholders.

optional 3rd party policy enforcement point

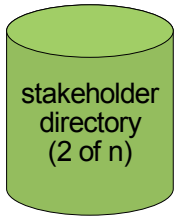
The repositories the directory must contact are defined within the authors' URIs. This has the side effect of defining the stakeholder's declaration of the repositories that hold its policy rulesheets.

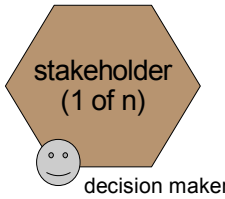


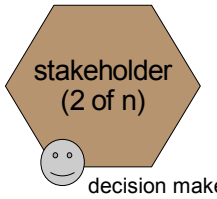


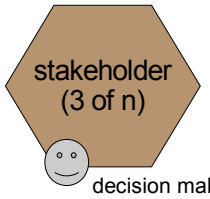


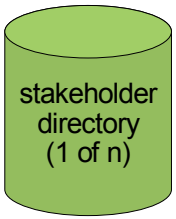
The author may cite various external dependencies within the rules, such as whether a license number exists in a given database. We'll see later how these external dependencies fit within the puzzle.





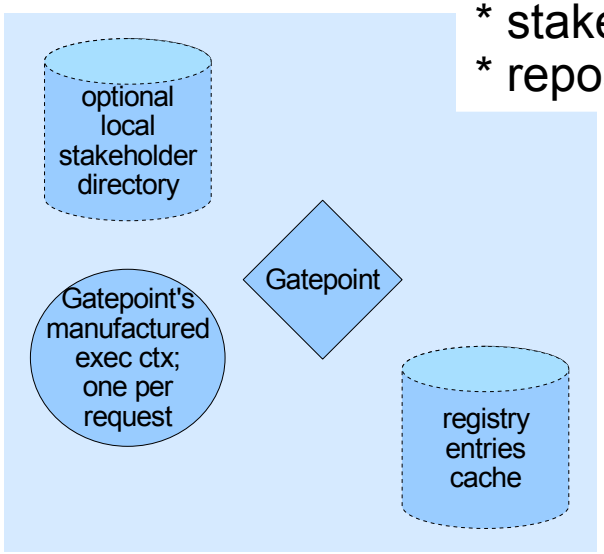


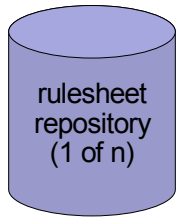


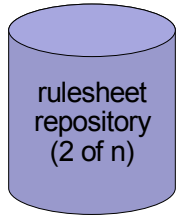


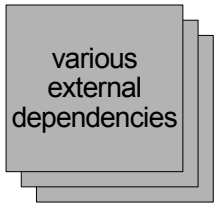
So, the stakeholder directories hold the keys to the CDCL operational kingdom. Within the directories is stored most information that defines how a rulesheet deck will get assembled:

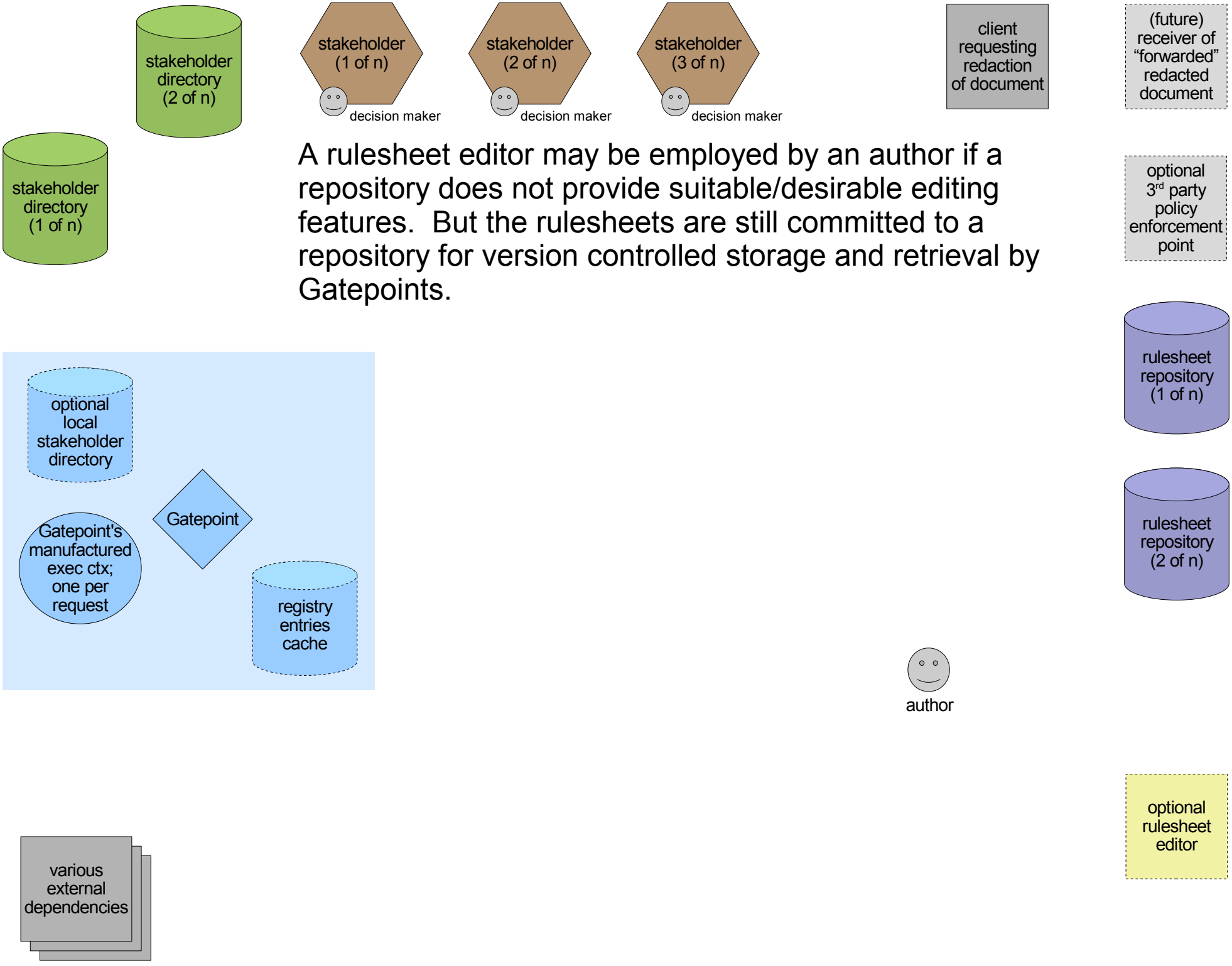
* stakeholder associations to other stakeholders
* stakeholder associations to document “problem spaces”
* stakeholder authorizations of rulesheet authors
* repositories' URIs that store stakeholders' policies

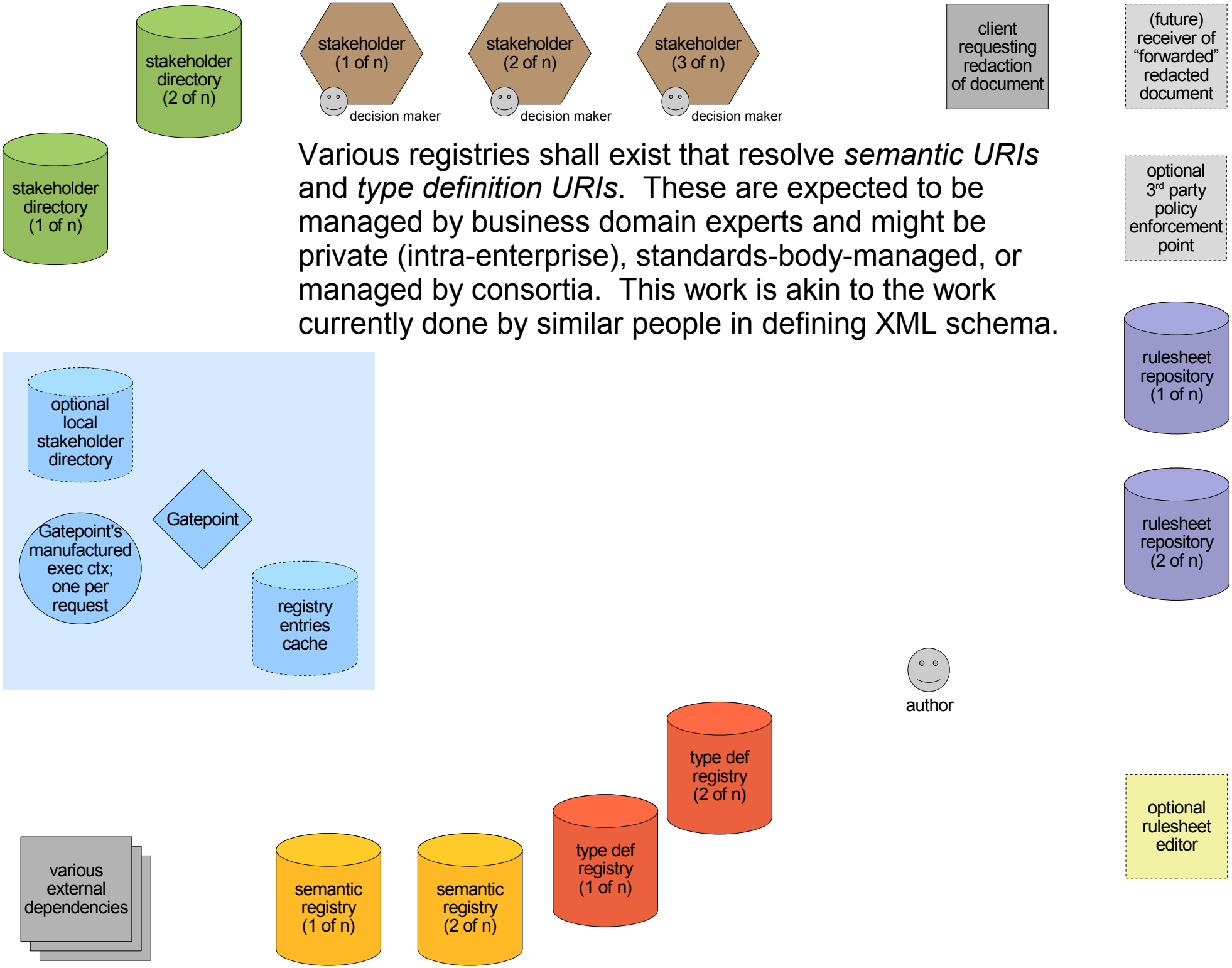


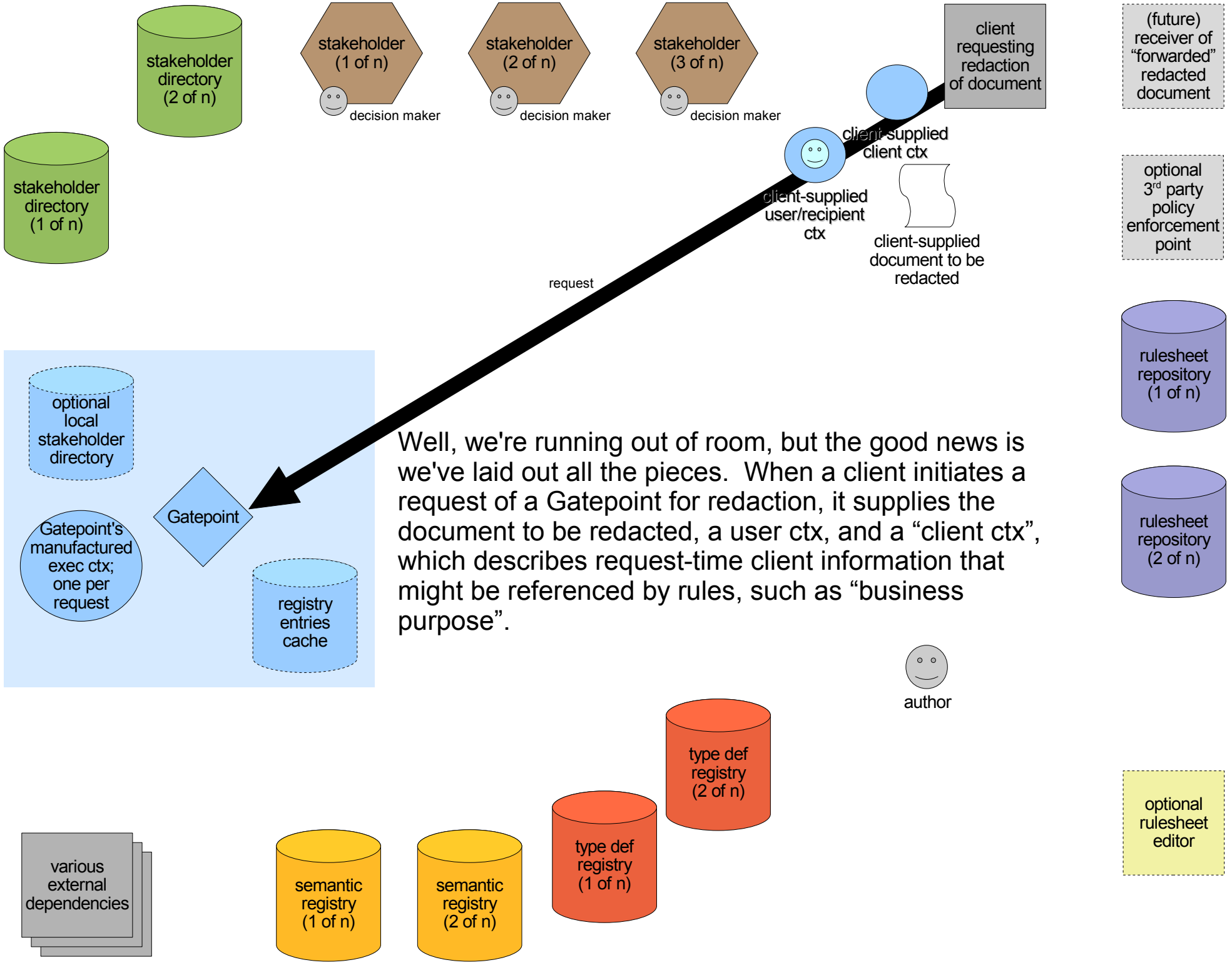


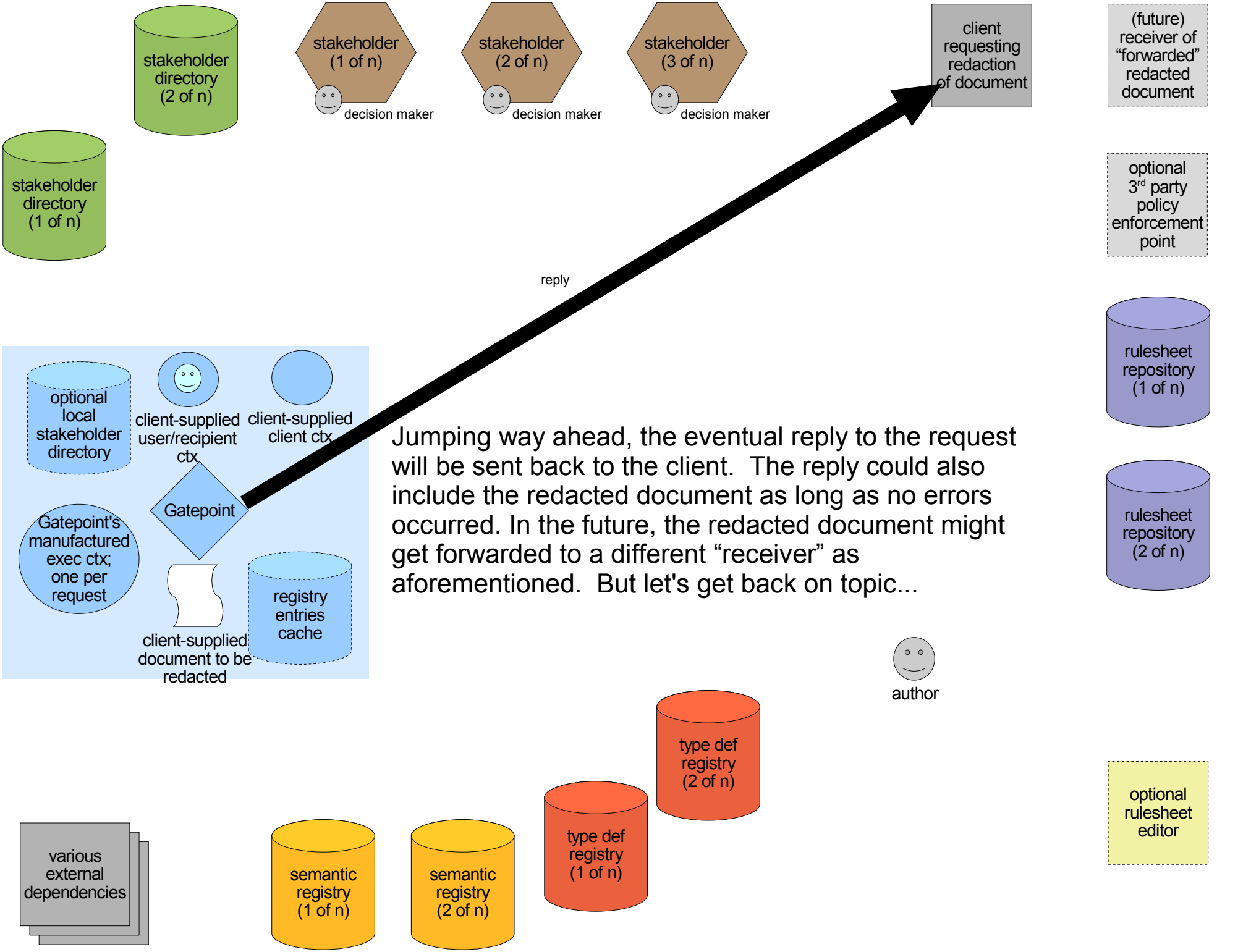


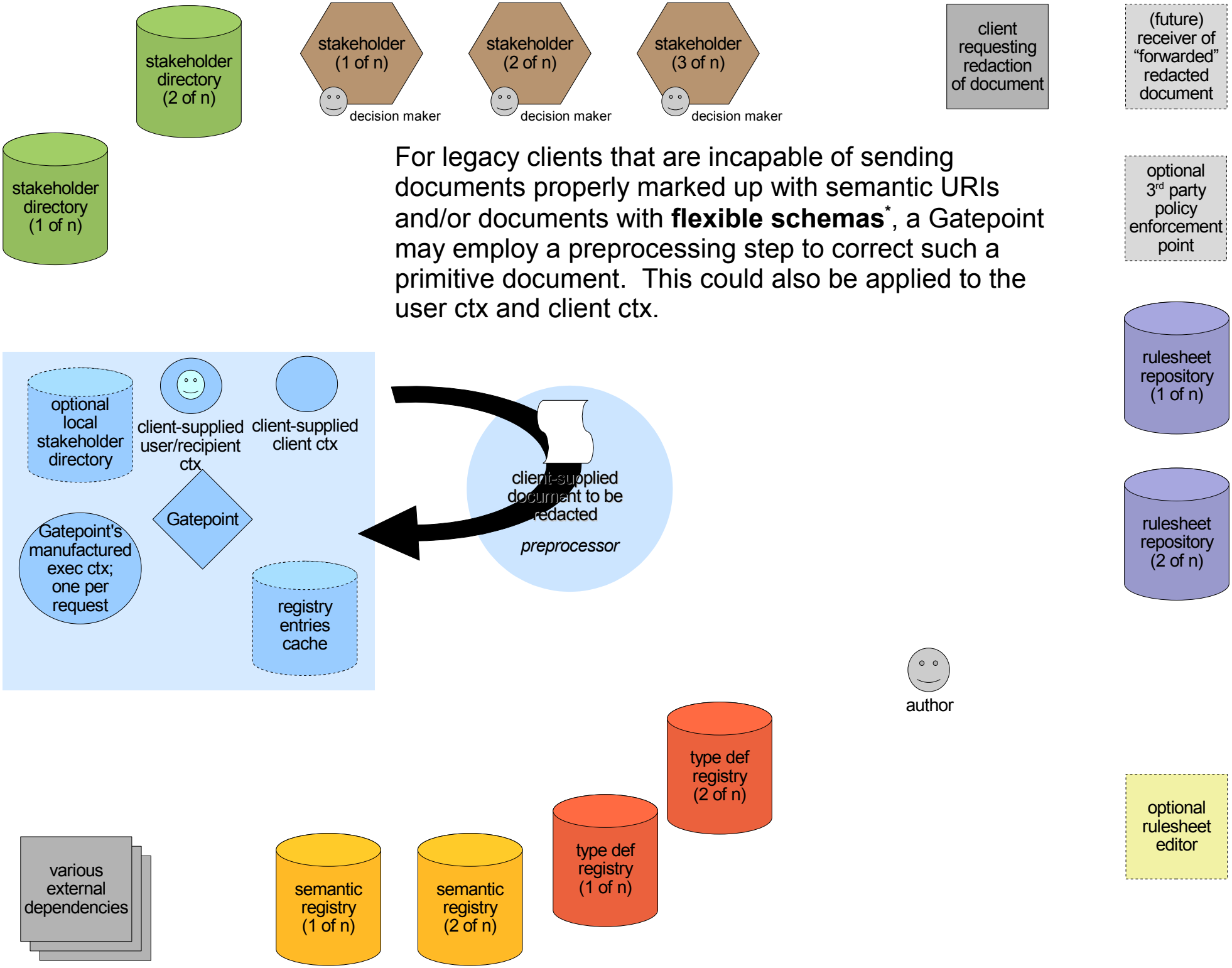


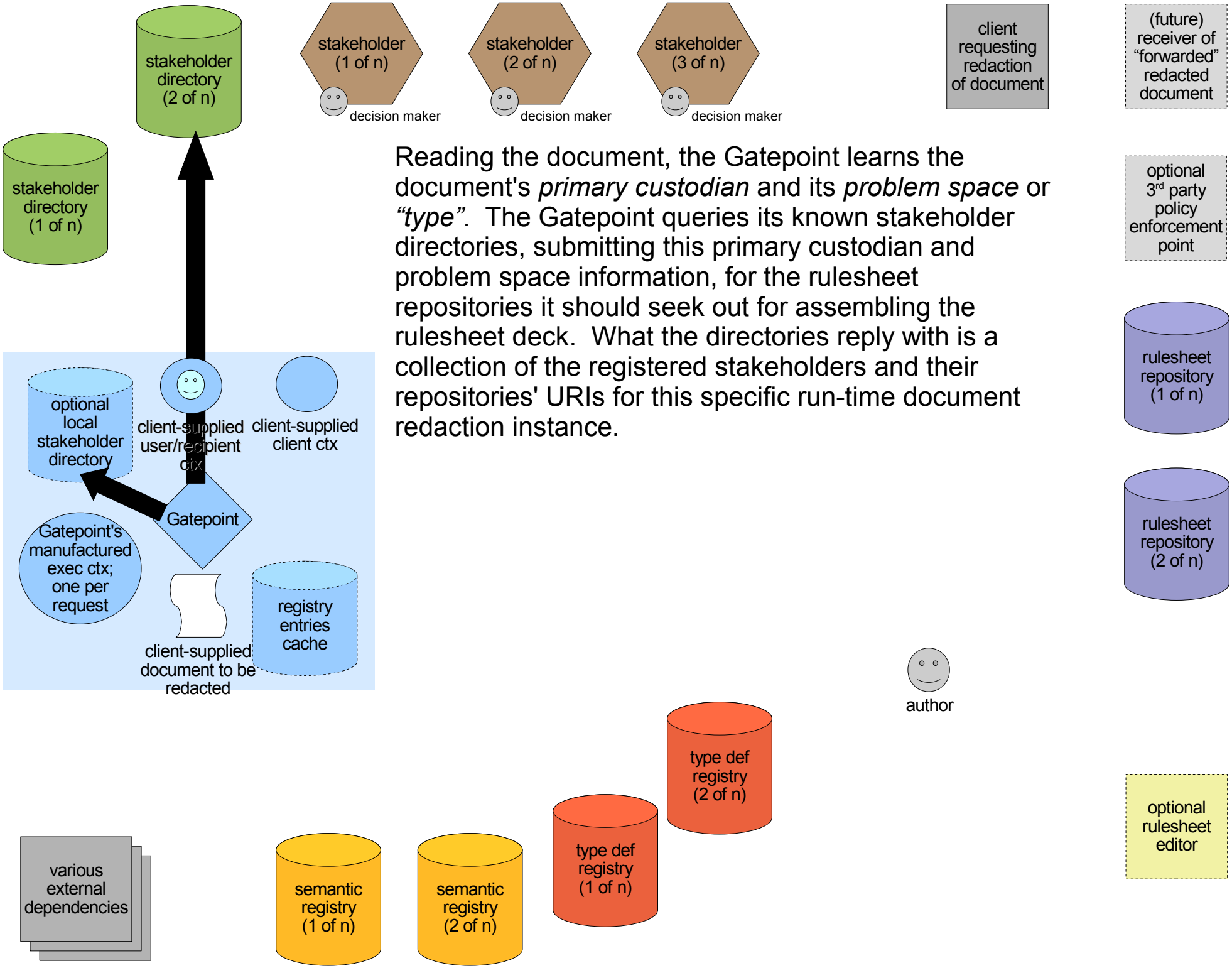

stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
A rulesheet editor may be employed by an author if a repository does not provide suitable/desirable editing features. But the rulesheets are still committed to a repository for version controlled storage and retrieval by Gatepoints.
optional 3rd party policy enforcement point
optional local stakeholder directory
Gatepoint
Gatepoint's manufactured exec ctx; one per request
registry entries cache
rulesheet repository (1 of n)
rulesheet repository (2 of n)
author
various external dependencies
optional rulesheet editor

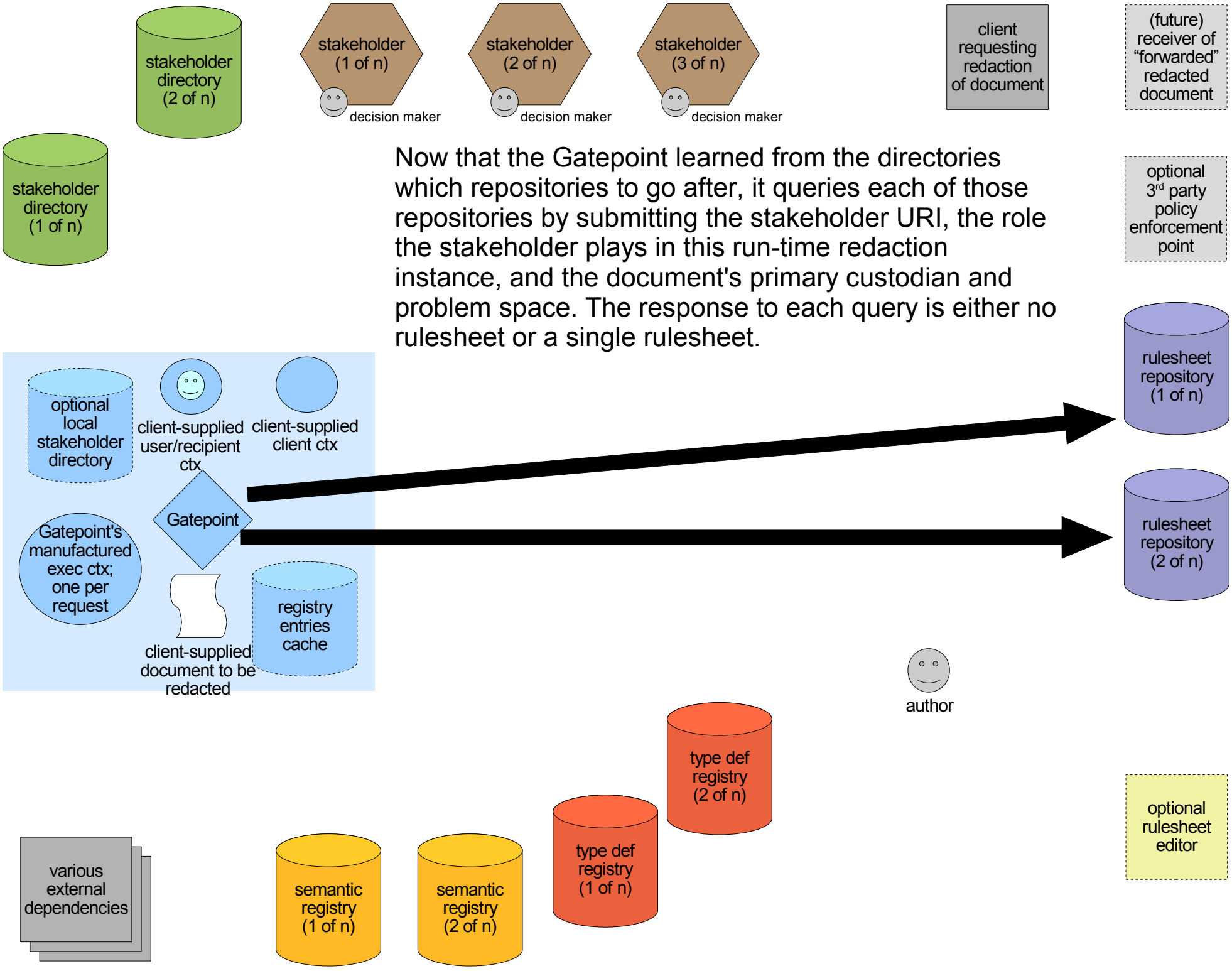

stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
optional 3rd party policy enforcement point
Various registries shall exist that resolve *semantic URIs* and *type definition URIs*. These are expected to be managed by business domain experts and might be private (intra-enterprise), standards-body-managed, or managed by consortia. This work is akin to the work currently done by similar people in defining XML schema.
rulesheet repository (1 of n)
optional local stakeholder directory
Gatepoint
Gatepoint's manufactured exec ctx; one per request
registry entries cache
rulesheet repository (2 of n)
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)

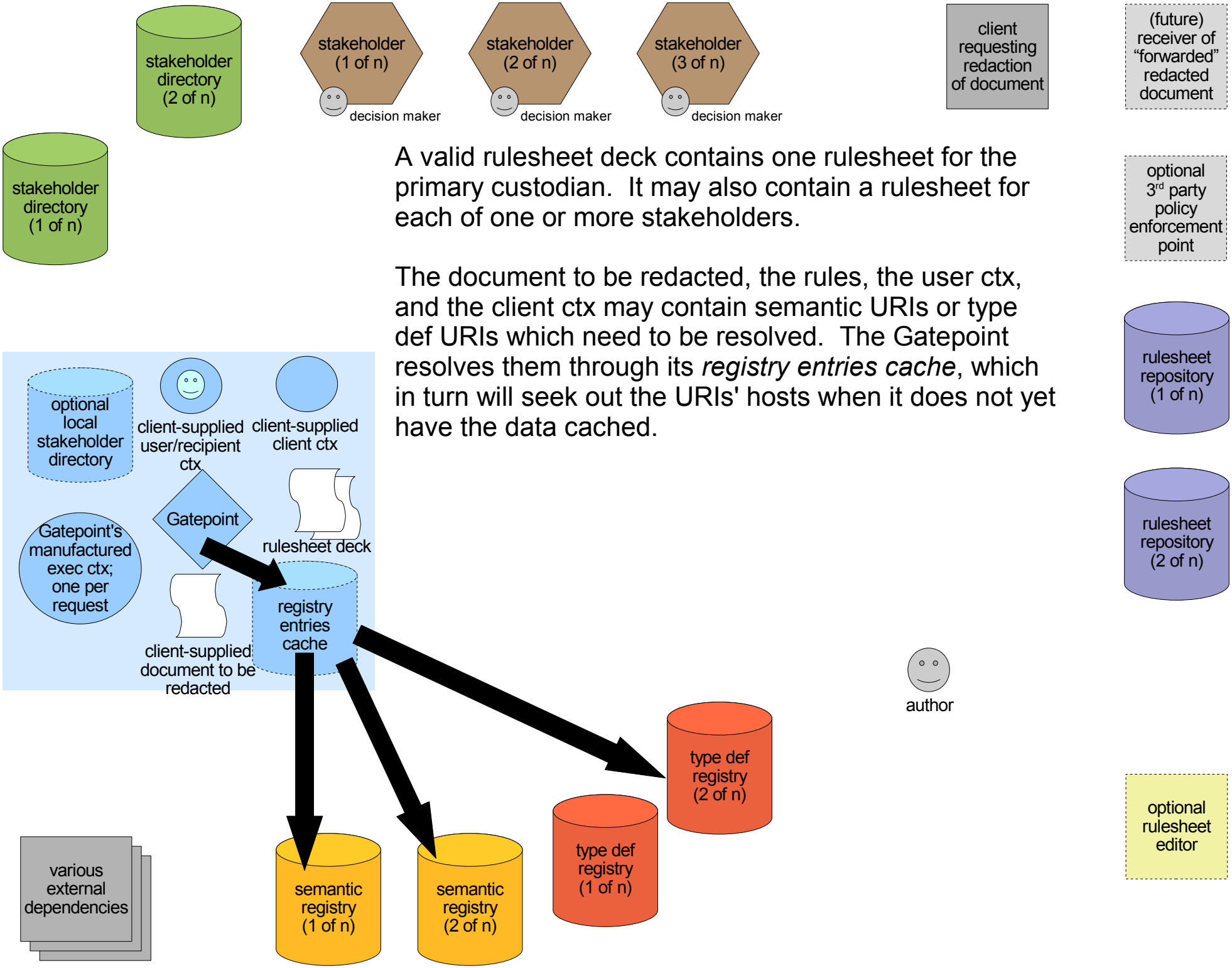

stakeholder directory (2 of n)
stakeholder (1 of n)
stakeholder (2 of n)
stakeholder (3 of n)
decision maker
decision maker
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
client-supplied client ctx
client-supplied user/recipient ctx
client-supplied document to be redacted
optional 3rd party policy enforcement point
request
rulesheet repository (1 of n)
optional local stakeholder directory
Gatepoint
Gatepoint's manufactured exec ctx; one per request
registry entries cache
rulesheet repository (2 of n)
Well, we're running out of room, but the good news is we've laid out all the pieces. When a client initiates a request of a Gatepoint for redaction, it supplies the document to be redacted, a user ctx, and a "client ctx", which describes request-time client information that might be referenced by rules, such as "business purpose".
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)

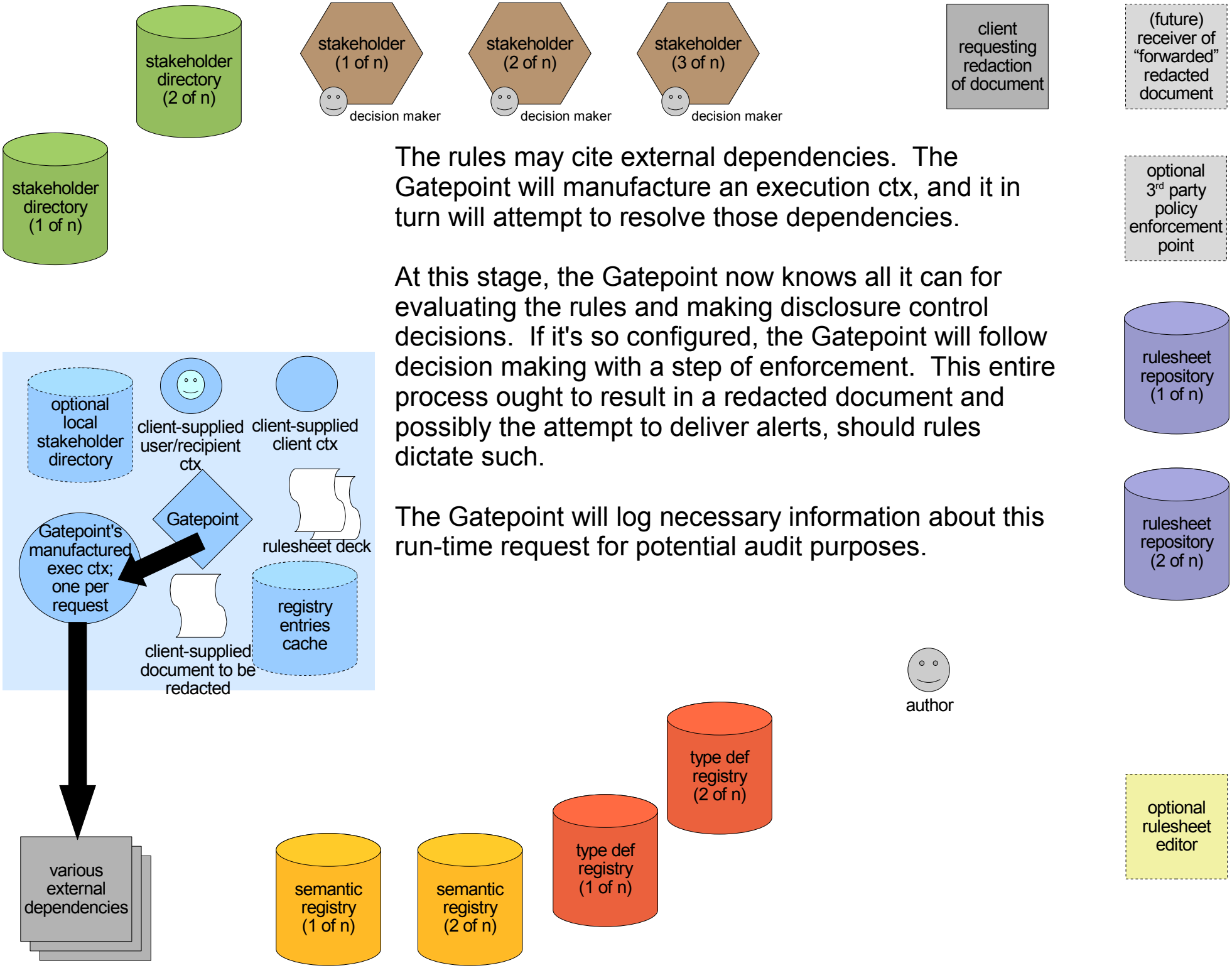

stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
optional 3rd party policy enforcement point
reply
optional local stakeholder directory
client-supplied user/recipient ctx
client-supplied client ctx
Gatepoint
Gatepoint's manufactured exec ctx; one per request
client-supplied document to be redacted
registry entries cache
rulesheet repository (1 of n)
rulesheet repository (2 of n)
Jumping way ahead, the eventual reply to the request will be sent back to the client. The reply could also include the redacted document as long as no errors occurred. In the future, the redacted document might get forwarded to a different "receiver" as aforementioned. But let's get back on topic...
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)

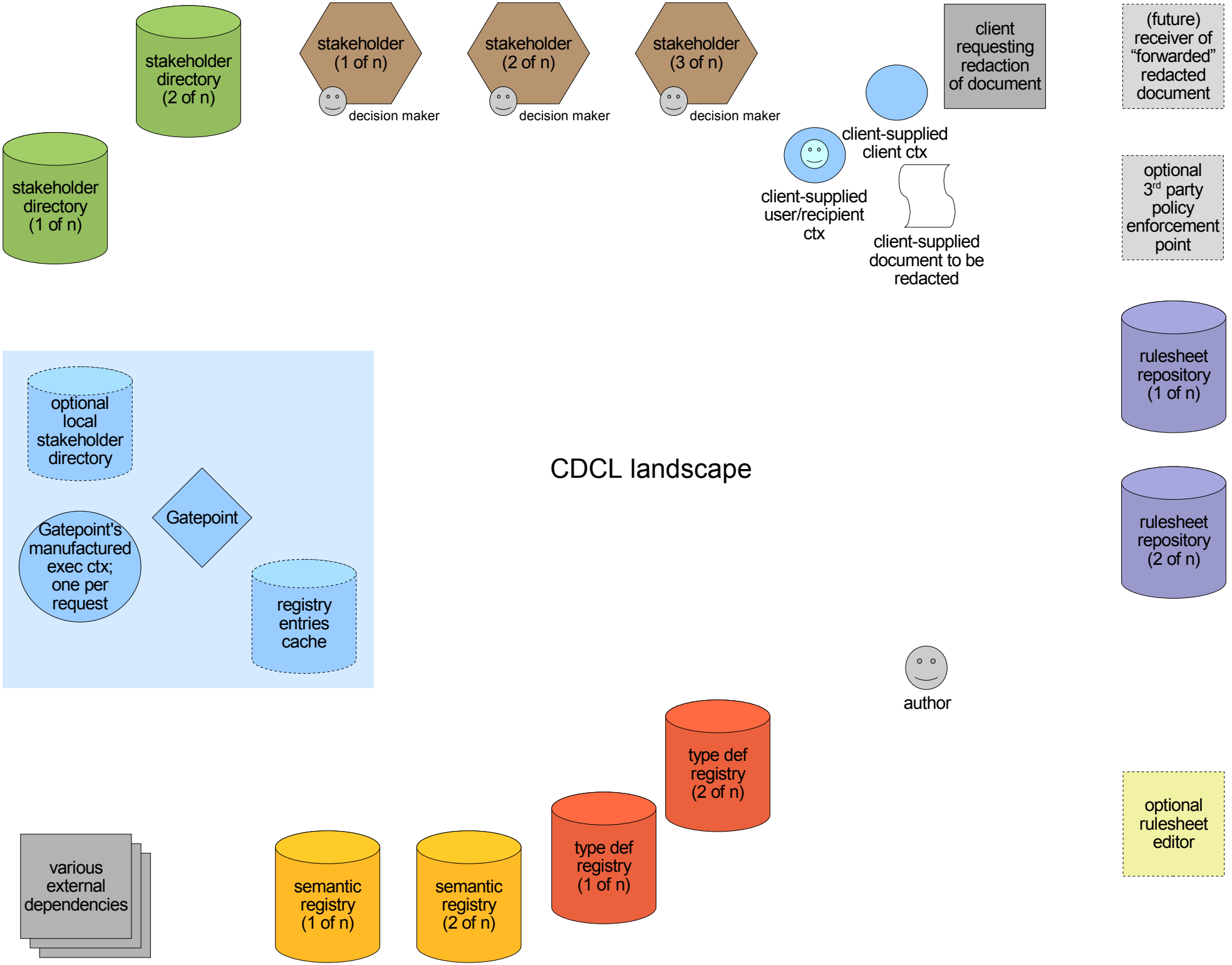

stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
optional 3rd party policy enforcement point
For legacy clients that are incapable of sending documents properly marked up with semantic URIs and/or documents with flexible schemas*, a Gatepoint may employ a preprocessing step to correct such a primitive document. This could also be applied to the user ctx and client ctx.
rulesheet repository (1 of n)
optional local stakeholder directory
client-supplied user/recipient ctx
client-supplied client ctx
Gatepoint
Gatepoint's manufactured exec ctx; one per request
registry entries cache
client-supplied document to be redacted
preprocessor
rulesheet repository (2 of n)
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)


stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
optional 3rd party policy enforcement point
Reading the document, the Gatepoint learns the document's *primary custodian* and its *problem space* or *"type"*. The Gatepoint queries its known stakeholder directories, submitting this primary custodian and problem space information, for the rulesheet repositories it should seek out for assembling the rulesheet deck. What the directories reply with is a collection of the registered stakeholders and their repositories' URIs for this specific run-time document redaction instance.
rulesheet repository (1 of n)
optional local stakeholder directory
client-supplied user/recipient ctx
client-supplied client ctx
Gatepoint
rulesheet repository (2 of n)
Gatepoint's manufactured exec ctx; one per request
registry entries cache
client-supplied document to be redacted
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)


stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
optional 3rd party policy enforcement point
Now that the Gatepoint learned from the directories which repositories to go after, it queries each of those repositories by submitting the stakeholder URI, the role the stakeholder plays in this run-time redaction instance, and the document's primary custodian and problem space. The response to each query is either no rulesheet or a single rulesheet.
optional local stakeholder directory
client-supplied user/recipient ctx
client-supplied client ctx
Gatepoint
Gatepoint's manufactured exec ctx; one per request
client-supplied document to be redacted
registry entries cache
rulesheet repository (1 of n)
rulesheet repository (2 of n)
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)


stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
optional 3rd party policy enforcement point
A valid rulesheet deck contains one rulesheet for the primary custodian. It may also contain a rulesheet for each of one or more stakeholders.
The document to be redacted, the rules, the user ctx, and the client ctx may contain semantic URIs or type def URIs which need to be resolved. The Gatepoint resolves them through its registry entries cache, which in turn will seek out the URIs' hosts when it does not yet have the data cached.
rulesheet repository (1 of n)
rulesheet repository (2 of n)
optional local stakeholder directory
client-supplied user/recipient ctx
client-supplied client ctx
Gatepoint
rulesheet deck
Gatepoint's manufactured exec ctx; one per request
client-supplied document to be redacted
registry entries cache
author
type def registry (2 of n)
type def registry (1 of n)
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)
optional rulesheet editor


stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
optional 3rd party policy enforcement point
The rules may cite external dependencies. The Gatepoint will manufacture an execution ctx, and it in turn will attempt to resolve those dependencies.
At this stage, the Gatepoint now knows all it can for evaluating the rules and making disclosure control decisions. If it's so configured, the Gatepoint will follow decision making with a step of enforcement. This entire process ought to result in a redacted document and possibly the attempt to deliver alerts, should rules dictate such.
The Gatepoint will log necessary information about this run-time request for potential audit purposes.
rulesheet repository (1 of n)
rulesheet repository (2 of n)
optional local stakeholder directory
client-supplied user/recipient ctx
client-supplied client ctx
Gatepoint
rulesheet deck
Gatepoint's manufactured exec ctx; one per request
registry entries cache
client-supplied document to be redacted
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)


stakeholder directory (2 of n)
stakeholder (1 of n)
decision maker
stakeholder (2 of n)
decision maker
stakeholder (3 of n)
decision maker
client requesting redaction of document
(future) receiver of "forwarded" redacted document
stakeholder directory (1 of n)
client-supplied client ctx
client-supplied user/recipient ctx
client-supplied document to be redacted
optional 3rd party policy enforcement point
rulesheet repository (1 of n)
optional local stakeholder directory
Gatepoint
Gatepoint's manufactured exec ctx; one per request
registry entries cache
CDCL landscape
rulesheet repository (2 of n)
author
type def registry (2 of n)
type def registry (1 of n)
optional rulesheet editor
various external dependencies
semantic registry (1 of n)
semantic registry (2 of n)