



Liberty Trust Models Guidelines

Version: 1.0

Editors:

John Linn, RSA Laboratories

Contributors:

Sharon Boeyen, Entrust

Gary Ellison, Sun Microsystems

Niina Karhuluoma, Nokia

William MacGregor, Schlumberger

Paul Madsen, Entrust

Senthil Sengodan, Entrust

Serge Shinkar, Communicator

Peter Thompson, IEEE-ISTO

Abstract:

This document is non-normative. Its purpose is to provide guidance on a variety of models that can be applied to establish trust among Liberty components, discussing their characteristics and implications. Its emphasis is on authentication and business relationships among components performing Liberty protocols, rather than on other components within supporting infrastructures. The discussion considers Liberty Phase 1 circle-of-trust environments as well as extended models appropriate to support the inter-identity provider interaction requirements established within Phase 2. Its intended audience includes designers of Liberty protocols and deployers of Liberty implementations.

Filename: liberty-trust-models-guidelines-v1.0.pdf

1 Notice

2 Copyright © 2003 America Online, Inc.; American Express Travel Related Services; Bank of America; Bell Canada;
3 Cingular Wireless; Cisco Systems, Inc.; Communicator, Inc.; Deloitte & Touche LLP; Earthlink, Inc.; Electronic
4 Data Systems, Inc.; Entrust, Inc.; Ericsson; Fidelity Investments; France Telecom; Gemplus; General Motors;
5 Hewlett-Packard Company; i2 Technologies, Inc.; Intuit Inc.; MasterCard International; NEC Corporation; Netegrity;
6 NeuStar; Nextel Communications; Nippon Telegraph and Telephone Corporation; Nokia Corporation; Novell, Inc.;
7 NTT DoCoMo, Inc.; OneName Corporation; Openwave Systems Inc.; PricewaterhouseCoopers LLP; Register.com;
8 Royal Mail; RSA Security Inc; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; SK Telecom; Sony
9 Corporation; Sun Microsystems, Inc.; Symlabs, Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa International;
10 Vodafone Group Plc; Wave Systems;. All rights reserved.

11 This specification document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to
12 use the document solely for the purpose of implementing the Specification. No rights are granted to prepare
13 derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other
14 uses must contact the Liberty Alliance to determine whether an appropriate license for such use is available.

15 Implementation of certain elements of this Specification may require licenses under third party intellectual property
16 rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are
17 not, and shall not be held responsible in any manner, for identifying or failing to identify any or all such third party
18 intellectual property rights. **This Specification is provided "AS IS", and no participant in the Liberty Alliance
19 makes any warranty of any kind, express or implied, including any implied warranties of merchantability,
20 non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementors
21 of this Specification are advised to review the Liberty Alliance Project's website (<http://www.projectliberty.org/>) for
22 information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance
23 Management Board.

24 Liberty Alliance Project
25 Licensing Administrator
26 c/o IEEE-ISTO
27 445 Hoes Lane
28 Piscataway, NJ 08855-1331, USA
29 info@projectliberty.org

30 **Contents**

31	1. Introduction	4
32	2. Definitions, Taxonomy, and Conceptual Processing Procedure	5
33	3. Pairwise Trust Model Examples	13
34	4. Brokered Trust Model Examples	14
35	5. Community Trust Model Examples	17
36	6. Comparison Among Models	20
37	7. Trust Establishment Mechanisms	21
38	8. Integrating Trust Establishment Infrastructures with Liberty	28
39	9. Metadata and Trust Discovery	29
40	References	30

41 **1. Introduction**

42 This specification is non-normative. Its purpose is to provide guidance on a variety of models that can be applied
43 to establish trust among Liberty components, discussing their characteristics and implications. Its emphasis is on
44 authentication and business relationships among components performing Liberty protocols, rather than on other com-
45 ponents within supporting infrastructures. The discussion considers Liberty Phase 1 circle-of-trust environments as
46 well as extended models appropriate to support the inter-identity provider interaction requirements established within
47 Phase 2. Its intended audience includes designers of Liberty protocols and deployers of Liberty implementations.

48 The models identified can be applied as parallel alternatives, and can be hybridized with one another. Through use of
49 different models, it is possible for a given entity to obtain trust in other entities through different means and to different
50 levels. While this document discusses and compares characteristics of the different models, it does not attempt to
51 specify a universal strength ordering among them.

52 The document's structure is as follows. Following this Introduction, [Section 2](#) presents a taxonomy to organize
53 discussion of different alternatives for trust establishment, defines relevant terms, and discusses a conceptual procedure
54 for trust-related processing. The next sections present examples of various models for establishing business trust
55 between Liberty entities:

- 56 • [Section 3](#) considers trust establishment on a pairwise basis, as is applied in Liberty's Phase 1 circles of trust.
- 57 • [Section 3](#) considers the use of active brokering entities as intermediaries to support transactions involving multiple
58 identity providers. This corresponds to the introducer model contemplated for support in Phase 2.
- 59 • [Section 5](#) considers interactions among Liberty components in a mode where interoperability is enabled through
60 the use of a common authentication infrastructure, and on business-level trust gained through that infrastructure's
61 administrative and enrollment processes, rather than on business agreements established independently of the
62 authentication infrastructure.

63 Within each of [Section 3](#) to [Section 5](#), alternative approaches for establishment of authentication trust are considered.
64 [Section 6](#) compares the presented models. [Section 7](#) provides a comparative overview of cryptographic trust
65 establishment methods, and [Section 8](#) discusses aspects of their application in the context of Liberty. [Section 9](#)
66 considers the prospect of metadata-based facilities for automated establishment of trust paths. [References](#) are the final
67 section.

68 **2. Definitions, Taxonomy, and Conceptual Processing Procedure**

69 This section defines relevant terms as used within this document, establishes a taxonomy to structure the discussion
70 of different trust model alternatives, and describes a conceptual processing procedure supporting the determination of
71 trust among communicating Liberty entities.

72 **2.1. Definitions**

73 *Authentication Enrollment Agreement* An agreement between an authentication infrastructure provider and an
74 entity registering in order to be authenticable through that provider's services. For
75 the case of PKI, where a CA acts as the infrastructure provider, provisions of an
76 authentication enrollment agreement will normally correspond to aspects of the CA's
77 applicable Certification Practice Statement (CPS).

78 *Brokered Trust* Brokered Trust describes the case where two entities do not have direct business
79 agreements with each other, but do have agreements with one or more intermediaries
80 so as to enable a business trust path to be constructed between the entities. The
81 intermediary brokers operate as active entities, and are invoked dynamically via protocol
82 facilities when new paths are to be established.

83 *Business Agreement* An agreement among parties providing the commercial prerequisites that the parties
84 require in order to engage in business transactions. Such agreements may be negotiated
85 bilaterally, or may be presented unilaterally by an issuer and accepted by a recipient.

86 *Business Anchor (BA)* A business anchor represents an entity with which its holder has a direct business
87 relationship. If an entity requires direct business agreements in order to interoperate
88 with other peers, those peers must be listed in the entity's business anchor list. If
89 an entity accepts indirect business agreements in order to interoperate with peers, its
90 business anchor list must identify an intermediary through which a business agreement
91 path can be derived leading towards those peers. A Business Anchor entry may be
92 qualified by the associated business agreement and other potential information such as
93 the subset of the TAL that applies to it.

94 *Business Anchor List (BAL)* Entities requiring business agreements in order to interoperate with other entities
95 will maintain business anchor lists identifying the entities with which direct business
96 trust relationships have been established. In some cases, these lists may correspond
97 with the trust anchor lists used to represent entities trusted for authentication purposes;
98 nonetheless, their semantics are distinct. Normally, entries in business anchor lists
99 will be added and removed only as a result of explicit administrative action, reflecting
100 changes to business agreements with direct partners.

101 *Community Trust* Community Trust applies when the business trust between a pair of entities is derived
102 from their enrollment in a common authentication infrastructure and acceptance of
103 its practices, without reliance on other business agreement paths. As such, the
104 entities' mutual trust in a business sense is based on their membership in a community
105 constructed and linked for authentication purposes.

106 *Direct Trust* Direct Trust is obtained when communicating entities hold each other's keys within their
107 TALs, so that their validity is established without reliance on intermediaries.

108 *Indirect Trust* Indirect Trust is obtained when communicating entities ascertain the validity of each
109 others' keys based on pre-existing trust established with an intermediary, as represented
110 by a trust anchor.

111	<i>Pairwise Trust</i>	Pairwise Trust describes the case where two entities have direct business agreements with each other.
112		
113	<i>Trust Anchor (TA)</i>	A trust anchor represents an entity and key that the anchor's holder has determined to trust directly for cryptographic authentication purposes. In some cases, the TA is qualified by an associated agreement between the represented entity and the TA's holder. This qualification may affect the set of entities that can be authenticated through the TA.
114		
115		
116		
117	<i>Trust Anchor List (TAL)</i>	Entities accepting cryptographic authentication of other entities will maintain trust anchor lists, identifying the entities and associated keys that they trust for authentication purposes and upon which validations will be based. In some cases, these lists may correspond with the business anchor lists used to represent entities trusted for business purposes; nonetheless, their semantics are distinct. Normally, entries in trust anchor lists will be added and removed only as a result of explicit administrative action reflecting changes in trust relationships.
118		
119		
120		
121		
122		
123		

124 **2.2. Taxonomy**

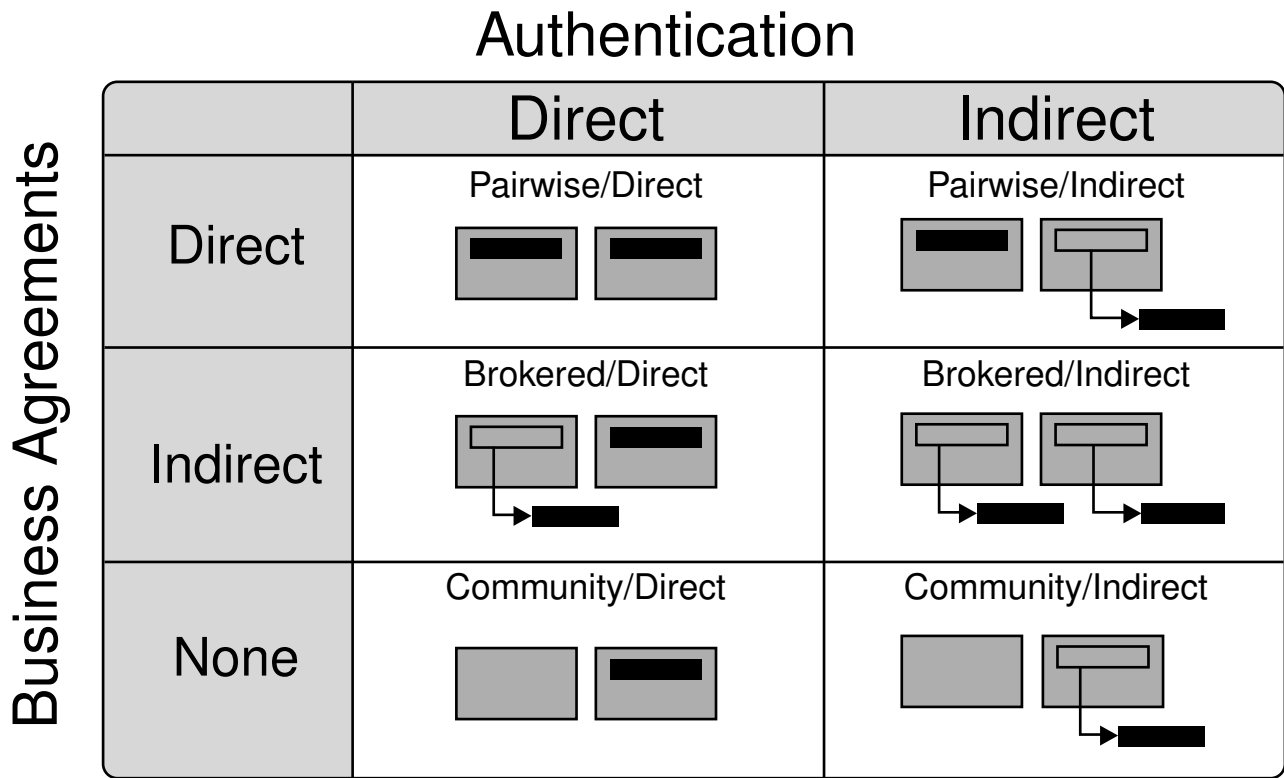
125 When issues of trust in distributed systems are discussed, confusion often results from ambiguities concerning
126 particular aspects for which entities are to be trusted. Figure 1 distinguishes two dimensions of trust, dimensions
127 introduced for clarification purposes.

128 The figure's columns distinguish the types of cryptographic infrastructures applied to support authentication among
129 components, ensuring that the identities of named entities are authentic. Proceeding along the horizontal axis,
130 we consider direct authentication (pairwise exchange of cryptographic keys), and indirect authentication (facilitated
131 through the involvement of off-line or on-line trusted intermediaries); since Liberty specifications require the use of
132 authentication facilities, no column is provided to represent unauthenticated cases. In the indirect case, it is common
133 for participants to accept authentication enrollment agreements issued unilaterally by the authentication infrastructure
134 providers; these help to ensure procedural integrity of the infrastructure, but are distinct from business-level agreements
135 executed between Liberty participant entities with the purpose of supporting Liberty-enabled services.

136 The figure's rows distinguish among the types of business agreements established between participants as a basis to
137 support transactions. Proceeding along the vertical axis, we consider direct agreements (exchanged between the
138 participants), indirect agreements (facilitated by business intermediaries), and the absence of business agreements
139 linking participants. Generally, it is assumed that business agreements will be negotiated between entities¹ on a
140 bilateral basis.¹

¹It has been suggested that certain intermediaries might provide unilateral business agreements to participants, facilitating establishment of indirect business agreement paths. This prospect requires further study, and may comprise a subcase of the Indirect Business Agreement table row.

141



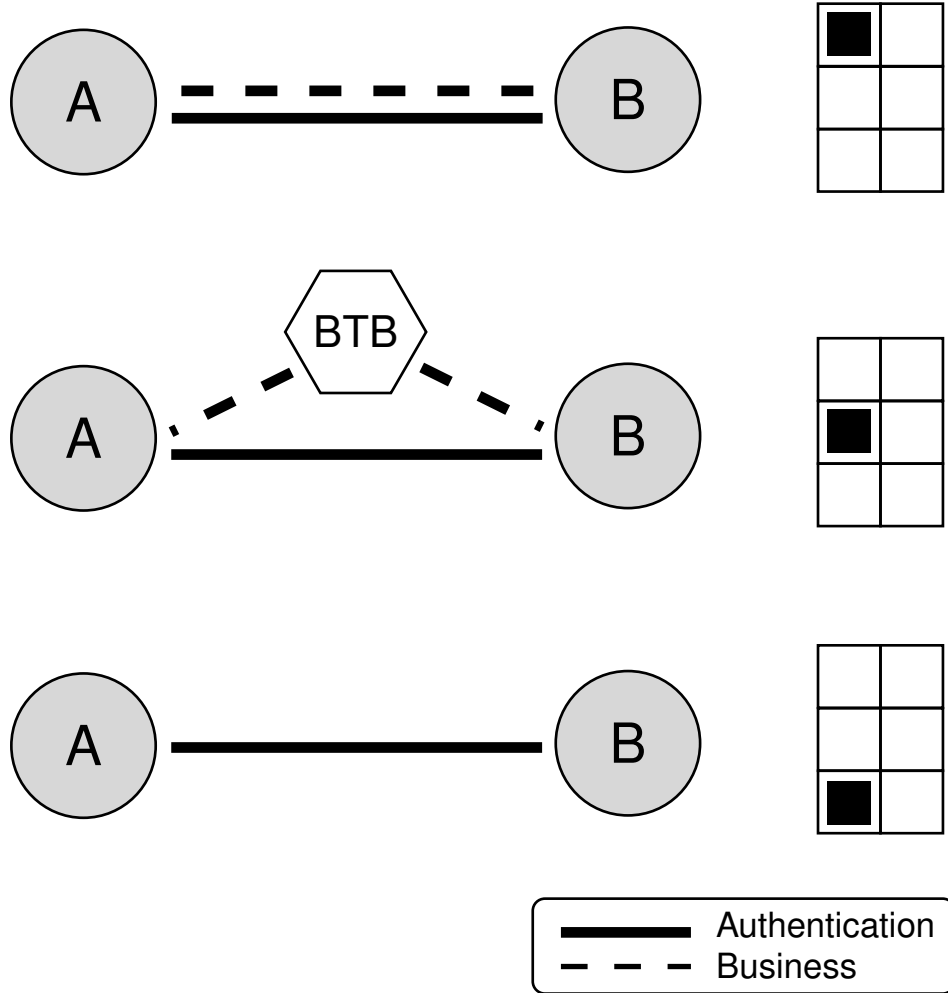
142

143

Figure 1. Trust Model Taxonomy

144 As the figure's structure suggests, approaches providing authenticated naming may vary independently from ap-
 145 proaches providing business-level trust. Titles within the figure's cells correspond to subsequent sections within
 146 the document, where supporting discussion will be provided. Within the cells, graphic elements represent applicable
 147 contents of the BAL (on left) and TAL (on right) corresponding to that case. In each graphic, the business entity in
 148 question is identified by a black horizontal rectangle. The cells indicate whether business agreement and authentication
 149 trust paths are direct, indirect, or absent using the following graphic conventions.

- 150 • For a direct path, by illustrating the black rectangle representing the business entity within either or both of the
 151 lists representing BAL and TAL,
- 152 • For an indirect path, by illustrating the black rectangle outside the applicable list but reachable through a link from
 153 some other entity (represented by a gray horizontal rectangle) located in the applicable list, or
- 154 • For an absent path, by the absence of a black rectangle or link thereto within the applicable list.

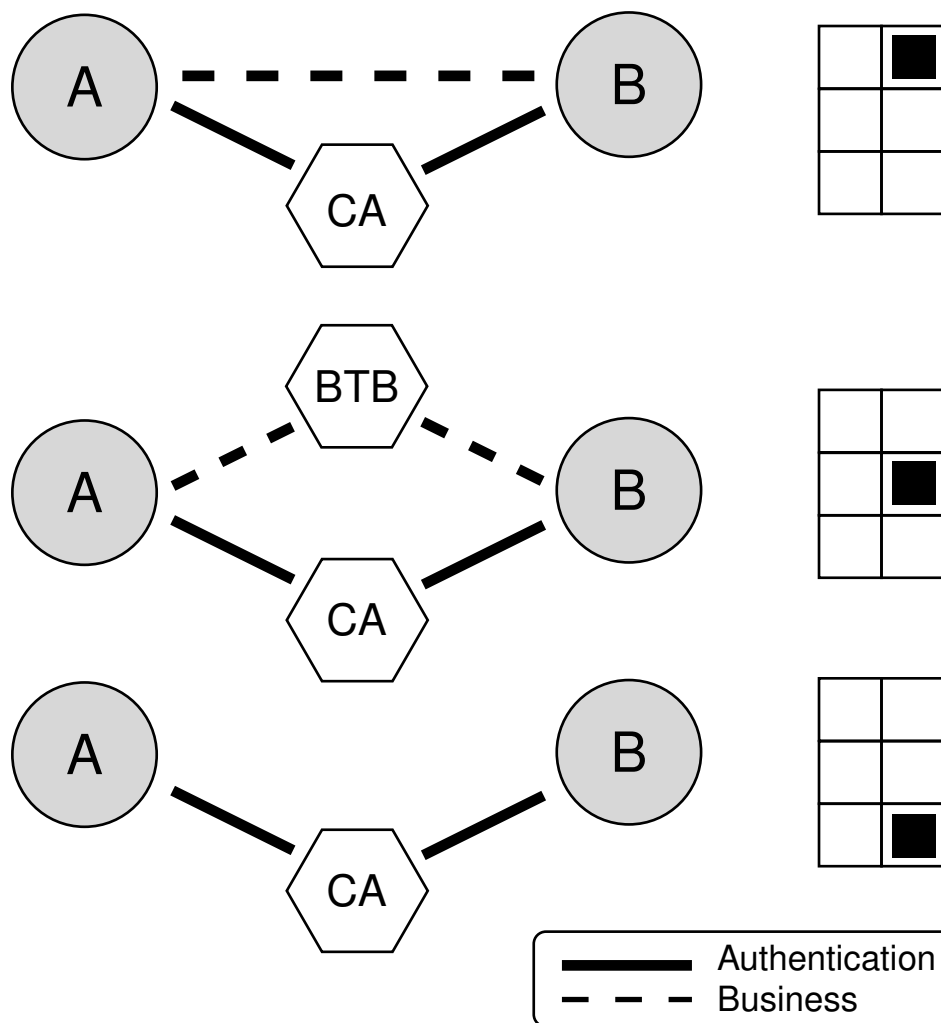


155

156

Figure 2. Direct Authentication Models

157 Figure 2 illustrates the three models based on direct authentication, associating them with their corresponding cells in
158 Figure 1.



159

160

Figure 3. Indirect Authentication Models

161 Figure 3 illustrates the three models based on indirect authentication (using a PKI CA as an example intermediary),
162 associating them with their corresponding cells in Figure 1.

163 2.2.1. Characteristics of Pairwise Trust Models

164 Liberty Phase 1 circles of trust exemplify Pairwise Trust models. These models afford strong trust in a business
165 sense, but have relatively limited scalability. Cryptographic authentication within these models may be based on
166 pairwise out-of-band exchange of shared secret keys or public-key certificates, in conjunction with business/legal
167 agreements; this exemplifies the Pairwise/Direct case. It is also possible for Phase 1 entities to authenticate each
168 other via an infrastructure involving intermediary entities (e.g., PKI CAs); such infrastructure usage exemplifies the
169 Pairwise/Indirect case.

170 In the Pairwise Trust models, relationship and business trust between all interoperating participants is exclusively
171 governed by signed business agreements. The strong trust established via business agreements is not technically
172 extendable which results in the forming of closed communities.

173 The determination of the level of trust in these communities is managed by business agreements, which generally
174 take precedence over trust established via authentication infrastructure. A new entity may not interact within such a
175 community without first entering into a business agreement with the existing participants and being added to the BAL.

176 **2.2.2. Characteristics of Brokered Trust Models**

177 In Liberty's Brokered Trust models, active intermediaries are invoked and involved when federation and/or authentication
178 transactions span multiple administrative domains. These approaches constrain the set of components that must
179 be involved in interdomain trust management, but require the use of additional protocol facilities beyond those defined
180 in Phase 1. Further, Brokered Trust models depend on availability of appropriate intermediaries in order to construct
181 a path to federate a user's relationship and/or to authenticate a particular session.

182 As an example situation Brokered Trust may be applicable, a service provider associated with identity provider A
183 receives an assertion to be processed from identity provider B, with which it shares no prior relationship. The
184 assertion may be an authentication assertion, a federation request, or an attribute assertion (in examples we will
185 refer to authentication assertion but it should be understood that this is merely representative of a more general
186 message). The service provider must decide whether to trust identity provider B's assertion. Overall trust is made up
187 of the combination of business trust, based on direct/indirect business agreements, and authentication trust, based on
188 direct/indirect cryptographic authentication infrastructure.

189 In Brokered Trust models, there is no direct business trust; i.e., the remote identity provider is not directly represented
190 in the BAL of the local service provider. However, there must be at least one entity represented in the local service
191 provider's BAL that can act as an intermediary for the local service provider. Two subcases are possible, depending
192 on the business agreements involved:

193 1. In the first subcase, it is assumed that the business agreement between the local service provider and the
194 intermediary explicitly identifies the remote identity provider as an entity with which the intermediary has a direct
195 business agreement and that this agreement can be used transitively with the agreement between the local service
196 provider and intermediary. This model enables the formation of a business agreement chain that satisfies the
197 business needs of the local service provider such that it may place trust in an assertion received from that remote
198 identity provider. No dynamic update protocol for the set of such remote entities per local business agreement is
199 anticipated. Requiring explicit identification of remote entities with which an intermediary has direct agreements
200 limits the length of possible chains of business agreements to two. If longer business agreement chains become
201 necessary, then some repository service would be required to enable identification of remote business agreements
202 that can be used as links in a path between two communicating entities.

203 2. In the second subcase, the business agreement between the local service provider and the intermediary places
204 broader trust in the intermediary, allowing it to act as an agent for the service provider and to establish paths
205 to other parties without requiring that those parties be identified in advance in the business agreement between
206 the local service provider and the intermediary. This subcase can allow business trust to be established more
207 dynamically and to a broader range of peers.

208 In some cases the establishment of indirect business trust with a remote entity will not require any additional anchors
209 to be added to the BAL. In these cases, an entity that is already represented in that list acts as the intermediary to
210 broker business trust with the remote entity. In other cases, if no such intermediary is listed in the local entity's BAL,
211 an additional anchor will need to be added. This additional anchor could be either another intermediary or a Liberty
212 provider directly (implying that subsequent transactions would be Pairwise Trust). It is assumed that the addition of
213 an entity to the BAL is a serious decision and is not undertaken without ensuring that the new entity is properly vetted
214 in accordance with security, operational, and business policies.

215 **2.2.3. Characteristics of Community Trust Models**

216 Community Trust models presume neither direct nor indirect business agreement paths between communicating
217 entities. Instead, they rely on shared membership in a community defined by a cryptographic trust establishment
218 infrastructure as a basis to enable communication between entities for purposes of federation and/or authentication.
219 Public Key Infrastructure (PKI), Kerberos realms and inter-realm relationships, and PGP webs of trust represent
220 examples of available trust establishment infrastructures. In these models, a trust establishment infrastructure is
221 used in lieu of direct business agreements or intermediary entities acting as trust brokers.

222 When Community Trust applies between a pair of entities, trust establishment is not based on identification of BAL
223 entries corresponding to the communicating peers. Instead, entries within the entities' TALs identify an authentication
224 trust path. Aspects of that authentication trust path are governed by the infrastructure's Authentication Enrollment
225 Agreements, and can be applied as a basis to achieve business-level trust.

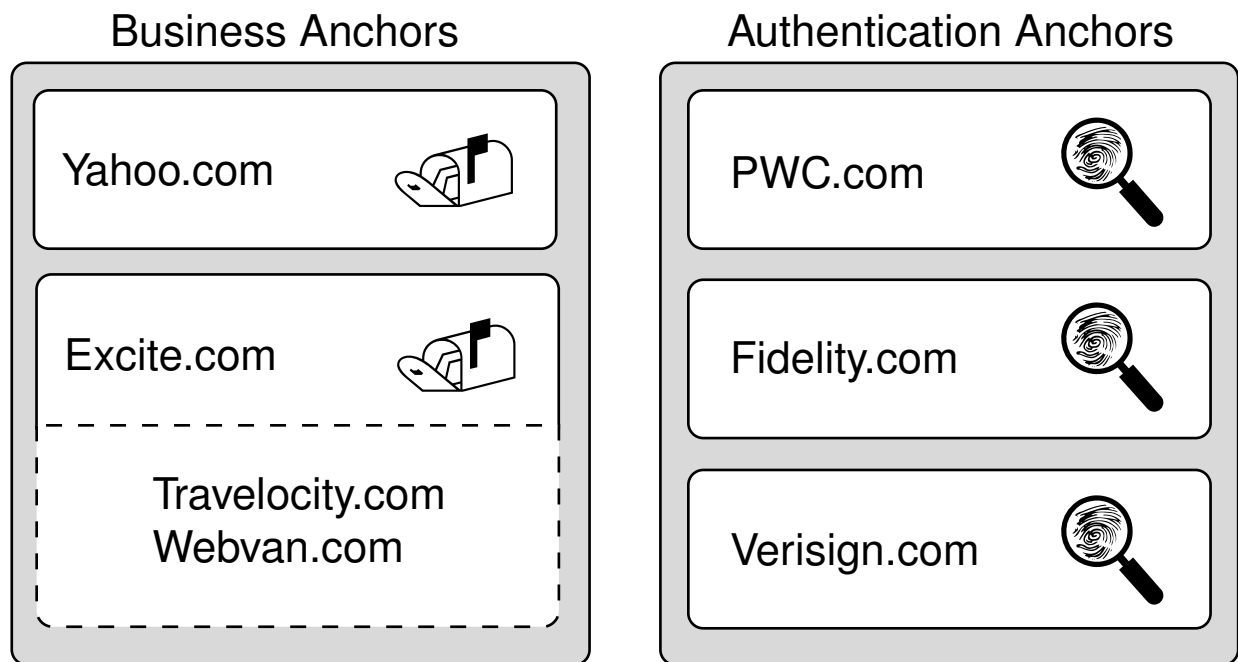
226 Hybrid models are also possible, where aspects of business-level trust obtained through the agreements of the Pair-
227 wise/Indirect or Brokered/Indirect models are complemented with additional aspects obtained through participation in
228 a common trust establishment infrastructure. Trust establishment infrastructures are essential to support these models
229 for authentication purposes, and can be leveraged to offer additional value for business purposes.

230 2.3. Conceptual Processing Procedure

231 For an entity A to determine whether a suitable basis exists to carry out trusted transactions with another entity B, it
232 operates on the following data:

- 233 • B's identity
- 234 • A's BAL
- 235 • A's TAL
- 236 • A's operational policies, indicating the types of paths it accepts

237 This section describes the necessary processing at a conceptual level; it is intended for descriptive purposes, not to
238 constrain individual implementations.



240 Figure 4. Example BAL and TAL

241 The process of validating an authentication trust path begins by determining whether A's TAL contains an entry for
242 B. If so (e.g., in the Figure 4 example, if B's identity is Fidelity.com), Direct Trust applies, and A possesses the
243 key required to authenticate messages and/or connections received from B. If not, A must determine whether one or
244 more of the entries in its TAL enables it to construct an authentication path to B. Path construction and validation
245 algorithms are well known, though their specifics vary for different types of infrastructures. If an authentication

246 path can be constructed and validated, Indirect Trust applies, and A can traverse that path to obtain the key required
247 to authenticate messages and/or connections received from B. If no path can be constructed, then A is unable to
248 authenticate B and the Liberty-specified prerequisites for communication cannot be satisfied. Assuming that A holds
249 or obtains the key necessary to authenticate B, it applies it as it processes B's communications, in order to validate B's
250 authenticity.

251 The process of validating a business agreement path begins by determining whether A's BAL contains an entry for B.
252 If so (e.g., in the Figure 4 example, if B is Yahoo.com), Pairwise Trust applies. If not, A must determine whether one
253 or more of the entries in its BAL enables it to construct a business agreement path to B. It appears that the process
254 of constructing business agreement paths has received less study in an algorithmic sense than that of constructing
255 authentication paths, so its procedures may often be more ad hoc in nature. If a business agreement path can be
256 constructed (e.g., in the Figure 4 example, a path to Travelocity.com via Excite.com), Brokered Trust applies. If not,
257 no business agreement applies between A and B, and any transactions must be carried out based on a Community
258 Trust model.

259 At this stage in the process, A has identified the "shortest" applicable type of authentication path (Direct or Indirect)
260 and of business agreement path (Pairwise, Brokered, or Community) reaching to B. It must now determine whether
261 these paths satisfy its policies and, if so, whether they dictate any limits or constraints on the transactions that it will
262 be willing to undertake with B; a peer reachable via Pairwise Trust, e.g, might be accorded broader rights than one
263 reachable only at the Community Trust level.

264 Note that some or all of A's BAL, TAL, and policy data may be kept confidential to A; it is not assumed that their
265 contents must be shared with B in order to enable transactions to proceed. It is possible, however, that sharing of some
266 of this information may simplify the task of identifying a suitable authentication and/or business agreement path.

267 **3. Pairwise Trust Model Examples**

268 **3.1. Pairwise/Direct Model**

269 In this model, an entity receives an assertion from another entity in its local circle of trust with which it has a direct
270 authentication trust established and business trust enabled. This direct authentication trust can be established by
271 exchanging keys using a means that is out-of-band with respect to Liberty specifications. The assertion recipient has
272 the assertion's originator in its TAL and BAL.

273 **3.1.1. Example**

274 As an example, a service provider signs a Business Agreement with an identity provider as part of which it agrees to
275 use the services of the identity provider to authenticate its users. The service provider adds the identity provider to its
276 BAL. The service provider and identity provider also set up a mechanism to exchange keys on a periodic basis. For
277 each period, the service provider picks up the key and stores the key in its TAL. The identity provider sends signed
278 assertions to the service provider, and the service provider uses the key it obtained in order to authenticate the identity
279 provider.

280 **3.2. Pairwise/Indirect Model**

281 In this model, an entity receives an assertion from another entity with which it does not have direct authentication trust
282 established. As such, the remote entity's key is not present in the local entity's TAL. The receiving entity does have a
283 Business Agreement with the sending entity and hence the sending entity is present in its BAL.

284 **3.2.1. Example**

285 Considering a PKI-based example, a service provider receives a signed authentication assertion from an identity
286 provider. Business trust exists between the two parties. If there is a valid certification path from one of the CA's
287 in the local service provider's TAL through a chain of intermediate CA's to the identity provider's certificate then the
288 signature on the assertion can be trusted.

289 **3.3. No Authentication Infrastructure**

290 This case is not conformant to Liberty specifications and is not recommended for operational use, but is described
291 briefly in the interests of clarification and completeness. Here, there exists no Authentication Infrastructure between
292 the service provider and identity provider but the identity provider and service provider have a business agreement.
293 This is likely to be a temporary state and not a likely permanent method unless one of the parties decides to forego
294 verification since it considers the services it provides of low value and not worth securing. This can occur temporarily
295 when existing infrastructure becomes unavailable due to it being compromised or broken. Hence the service provider
296 will not be able to authenticate the identity provider and will not be able to validate the assertions. The service
297 provider may determine that such an assertion can be used to provide service as the level it would be offered to users
298 anonymously or with unsigned authentication assertions from an identity provider.

299 **4. Brokered Trust Model Examples**

300 Each of the following subclauses describes a distinct model for authentication trust that is used in conjunction with
301 indirect business trust. These authentication trust models include direct authentication trust, indirect authentication
302 trust and no authentication trust.

303 **4.1. Brokered/Direct Model**

304 In this model, the local entity that receives an assertion from a remote entity has direct authentication trust established
305 with that remote entity. As such, the remote entity's key is included in the local entity's TAL. Because this model deals
306 with indirect business trust, the remote entity is not represented in the local entity's BAL.

307 **4.1.1. Example**

308 Considering an example, a local service provider receives a signed authentication assertion from a remote identity
309 provider. The local service provider has a local identity provider in its BAL. The business agreement between these
310 two does not explicitly state that the local identity provider has a business agreement with the remote identity provider.
311 The local identity provider provides business trust only among the service providers with which it is affiliated. Another
312 identity provider does have a business agreement with the remote identity provider and offers to act as an intermediary
313 for the local service provider. Such an identity provider may have as its primary role that of an intermediary broker.
314 Many Liberty entities could make use of such intermediaries to establish business agreement chains with remote
315 entities. Because of the generic nature that such business agreements would likely have, it may be that the services of
316 such brokers would be used primarily for lower value business transactions than those where a local identity provider
317 is used as the intermediary for business trust. The indirect business agreement chain includes the business agreement
318 between the local service provider and generic remote identity provider broker, as well as the business agreement
319 between the remote identity provider broker and the remote identity provider that initiated the authentication assertion.

320 Because the local service provider already has the key of the remote identity provider that issued the authentication
321 assertion in its TAL, no intermediary is required for cryptographic authentication trust.

322 The service provider has established indirect business trust and direct authentication trust. Together these enable
323 overall trust to be placed in the authentication assertion received from the remote identity provider. In this example, an
324 additional business anchor for the generic remote identity provider broker must be added to the local service provider's
325 BAL. No new trust anchors need to be added to its TAL.

326 **4.2. Brokered/Indirect Model**

327 In this model, the local entity that receives an assertion from a remote entity does not have direct authentication trust
328 established with that remote entity. As such, the remote entity's key is not present in the local entity's TAL. Because
329 this model deals with indirect business trust, the remote entity is also not represented in the local entity's BAL. The
330 examples vary in the authentication technologies they employ, and in whether their infrastructure components are
331 involved actively or passively in the authentication process. They include a PKI case, a Kerberos case, and a case
332 where SAML assertions are used as a basis for establishment of trust in a remote identity provider.

333 **4.2.1. Example 1: PKI**

334 To facilitate comparison of the examples in [Section 4.2.1](#) to [Section 4.2.3](#), the same basic scenario is used. A local
335 service provider receives a signed authentication assertion from a remote identity provider. In this example, indirect
336 business trust is established using one of the techniques described in the previous section.

337 Public-key infrastructure (PKI) is the authentication infrastructure in this example. The local service provider has in
338 its TAL the key of the CA that issued a public-key certificate used to verify the digital signature of the local identity
339 provider. If this same CA issued a certificate to the remote identity provider, then the signature on the authentication
340 assertion issued by the remote identity provider can be verified using that same trust anchor. Even if the same CA did
341 not issue a certificate to the remote identity provider, if there is a valid certification path from the local trust anchor,

342 through one or more intermediate CAs, to the certificate issued by some other CA to the remote identity provider, the
343 signature on the authentication assertion can be trusted.

344 The service provider has established indirect business trust and indirect authentication trust. Together these enable
345 overall trust to be placed in the authentication assertion received from the remote identity provider. Depending on
346 whether indirect business trust was established as in example 1 or example 2 of 4.1, the service provider may/may not
347 need to add a new anchor to its BAL. Because one of the CAs whose key is already in the local service provider's TAL
348 either issued a certificate directly to the remote identity provider or issued a certificate to an intermediary CA that is
349 used to form a valid certification path to the remote identity provider, no new anchor needs to be added to the local
350 service provider's TAL.

351 **4.2.2. Example 2: Kerberos**

352 As with the previous example, a local service provider receives a signed authentication assertion from a remote identity
353 provider. In this example, indirect business trust is established using one of the techniques described in the examples
354 in 4.1.

355 Kerberos is the indirect authentication infrastructure in this example. The local service provider's TAL contains the
356 symmetric key that it shares with its local KDC but does not contain a symmetric key for the remote identity provider.
357 In order for the local service provider to place authentication trust in the signed (HMACed) assertion from the remote
358 identity provider; that remote identity provider will have to demonstrate that it was trusted (directly - if it shares the
359 KDC with the local service provider or indirectly - if it belongs to another Kerberos realm). The remote identity
360 provider is able to demonstrate this trust by proving that it has possession of a short-lived symmetric key that was also
361 delivered to the remote service provider encrypted by the long-lived symmetric key shared between the local service
362 provider and its KDC.

363 The service provider has established indirect business trust and indirect authentication trust. Together these enable
364 overall trust to be placed in the authentication assertion received from the remote identity provider. Depending on
365 whether indirect business trust was established as in example 1 or example 2 of 4.1, the service provider may/may
366 not need to add a new anchor to its BAL. If inter-realm Ticket-Granting Tickets (TGTs) traversing the path from the
367 remote identity provider's KDC to the local service provider's KDC are obtained and used, the local service provider
368 can authenticate the remote service provider's communications without adding a new TA to its TAL.

369 **4.2.3. Example 3: SAML**

370 Just as SAML Authentication Assertions enable indirect authentication trust between Principals and service providers
371 (with the identity provider playing the role of TTP), SAML can play a similar role enabling indirect authentication
372 trust between local service providers and remote identity providers.

373 Logically very similar to the Kerberos example above, the local service provider will be able to derive trust in the
374 remote identity provider through the active involvement of a TTP playing the logical role of the Kerberos KDC, i.e.
375 issuing authentication tokens to the remote identity provider that will be trusted by the local service provider because
376 of the trust the service provider has in the TTP. While in the previous example these authentication tokens are binary
377 Kerberos tickets, in this example they are SAML Authentication Assertions.

378 The local service provider's TAL either directly contains the public key of the TTP or contains the key of a CA that
379 has issued a certificate to that TTP such that the service provider can verify SAML Authentication Assertions signed
380 by the TTP's associated private key. By definition, the local service provider's TAL does not contain a key for the
381 remote identity provider.

382 The remote identity provider authenticates to the TTP (SAML Authentication Authority) in order to be issued a
383 SAML Authentication Assertion, signed by the TTP. The remote identity provider then presents the SAML assertion
384 as a 'letter of introduction' to the local service provider. The SAML Authentication Assertion will likely contain
385 keying information encrypted for the local service provider. The remote identity provider is able to demonstrate its
386 trustworthiness to the remote service provider by proving that it has possession of the same key. This shared secret

387 will allow the remote identity provider and the service provider to securely establish a session key for their subsequent
388 transaction. Following completion of this processing, the service provider has established indirect business trust and
389 indirect authentication trust. Together, these enable overall trust to be placed in the authentication assertion received
390 from the remote identity provider.

391 Like the Kerberos example, this use of SAML relies on a TTP playing an active role in the derivation of indirect trust
392 through the real-time issuance of authentication tokens. Unlike the Kerberos example, this SAML scenario depends
393 on asymmetric cryptography. The authenticity of the SAML Authentication Assertions is determined by private key
394 signatures rather than a secret key MAC.

395 **4.3. No Authentication Infrastructure**

396 This case is not conformant to Liberty specifications and is not recommended for operational use, but is described
397 briefly in the interests of clarification and completeness. In some situations, an entity in one domain may need
398 to establish trust with an entity in another domain, even though there is no supporting cryptographic authentication
399 infrastructure (direct or indirect) in place. For example, in a situation where one company purchases another, the
400 subsumed organization may inherit the business agreements of the parent company but not yet have cryptographic
401 authentication infrastructure established to support those business agreements. Given the same scenario as above,
402 where a service provider in the subsumed company receives a signed authentication assertion from an identity provider
403 in another domain, the service provider may be able to establish indirect business trust, but no authentication trust. As
404 such, the local service provider may still be able to use that authentication assertion, although the level of overall trust
405 in that assertion would be reduced. The local service provider may determine that such an assertion can be used to
406 provide service as the level it would be offered to users anonymously or with unsigned authentication assertions from
407 an identity provider.

408 **5. Community Trust Model Examples**

409 In the Community Trust model, an organization (e.g., an industry consortium or a community) sponsors, endorses, or
410 adopts one or more trust establishment services to provide and manage the credentials needed by entities to create and
411 maintain authentication trust among themselves. The service(s) could be operated by the sponsoring organization, or
412 could be provided by an independent service delivery organization. In Community Trust, some level of business trust,
413 although not provided by either direct or brokered business agreements, can be derived from participation in a shared
414 authentication infrastructure. The assumption is that the authentication infrastructure will, in addition to allowing
415 entities to be identified, further identify them as belonging to some community.

416 Various service options are possible; with PKI technology, e.g., the set of selected services could include one, some,
417 or all of:

- 418 • Certification Authorities (CAs);
- 419 • Publication repositories for certificates and CRLs, whether generated by sponsored services or obtained from other
420 sources (e.g., from independent CAs maintained by participants rather than a community-level facility);
- 421 • On-line facilities for certificate status checking.

422 Different options imply different degrees of organizational involvement and, potentially, of organizational liability.
423 Generally, a broader set of services will incur greater costs than a narrower set, but will also afford more value in terms
424 of enabling trusted connectivity among participant entities and of ensuring consistent assurance across the participant
425 community.

426 **5.1. Community/Direct Model**

427 The simplest cases of direct authentication involve small configurations and manual keying, and a privileged officer
428 responsible for all key management actions. Direct authentication becomes unwieldy as the number of managed
429 entities grows, and consolidated repositories of key material, especially symmetric key material, can create a significant
430 security risk.

431 Considering an example, a small, multi-site, hub-and-spoke Liberty community agrees to rely on the direct exchange
432 of self-signed certificates to establish communications and authentication trust. Participants accord each other
433 community-level business trust based on their enrollment in this process. The operator of each entity has a software
434 tool that will create PKI key pairs and create self-signed X.509v3 certificates. The identity provider operator creates
435 two key pairs, one for SSL/TLS and one for XML-Signature use, and delivers the corresponding certificates to each of
436 the service provider operators in a secure manner (e.g., by personal meeting, or by email and subsequent out-of-band
437 verification of the certificate fingerprints). Each service provider operator creates one key pair for XML-Signature
438 use, and delivers the corresponding certificate to the identity provider operator in a secure manner.

439 This example uses the technical mechanisms of PKI, in the form of asymmetric key pairs and certificates, without
440 reliance on a Trusted Third Party or Certification Authority. It is therefore an intermediate step, benefiting from
441 ubiquitous technology but not leveraging the advantages of an available TTP service. This approach can be used
442 effectively, but has three major drawbacks:

- 443 1. without the stabilizing effect of a TTP and its policies, the necessary discipline and rigor for trusted operation is
444 easily lost (e.g., certificates are exchanged via email but the fingerprint verification may never be done);
- 445 2. the trust establishment process is straightforward, but trust disestablishment, when a service provider operator
446 goes out of business, for example, requires extreme diligence among participants; and
- 447 3. each party assumes full responsibility for identity verification of the other parties.

448 **5.2. Community/Indirect Model**

449 Indirect authentication implies the use of trust infrastructure services outside of the Liberty model. Available trust
450 establishment services can improve the assurance level of Liberty operations, and/or reduce the cost of operations,
451 because they potentially deliver identity verification, credential lifecycle management, and credit checks and other
452 qualification ratings, obtained under well-defined, implemented, and audited policies and procedures. These aspects
453 can be important in the acceptance of corresponding community-level trust relationships for business purposes. Under
454 the assumption that a trust infrastructure service is already available and the participating entities are already enrolled
455 in the infrastructure in other capacities, use of an available trust infrastructure service may also avoid duplication of
456 effort.

457 **5.2.1. Example: PKI Certification Hierarchies**

458 Considering one example, a Liberty community agrees on a list of TTPs offering PKI certificate services. In
459 addition to conventional Certification Authorities (CAs), Bridging Authorities may also be included. In the latter
460 case, each Bridging Authority cross certifies with participating CAs and with other Bridging Authorities. Two types
461 of approaches can be applied (or hybridized) to establish trust among community members:

- 462 • Individual entities' trusted CAs establish cross-certification paths to other CAs within the community, and the
463 entities employ their existing trust anchors that reference their trusted CAs;
- 464 • A list of selected trust roots representing the set of the Community's CAs becomes a Community TAL. This TAL
465 is distributed to all of the entities in the community in a secure manner.

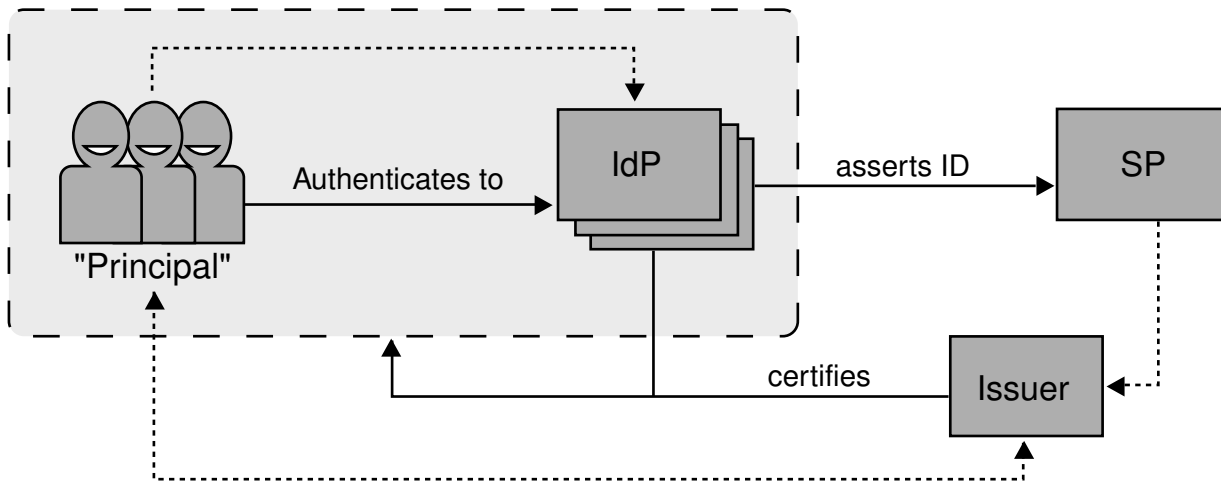
466 For each trust root there is a certificate verification procedure known to the participating entities. Given any certificate,
467 an entity can apply the certificate verification procedures, and positively determine if the certificate in question was
468 issued in accordance with the policies of one or more of the TTPs trusted by the community. The entity can also
469 determine, according to the policies of the TTPs, if the certificate is still valid (i.e., has not expired, and has not been
470 revoked).

471 This example represents a "full PKI" case. The selected TTPs may be commercial, government operated, or
472 closed community service providers, and the TAL creates flexibility to adjust the mix over time. As a matter of
473 community policy, the trust anchors could be required to share a single certificate verification procedure, simplifying
474 the implementations of the participating entities; or multiple procedures could be allowed to increase the pre-enrolled
475 population or enable technology migration.

476 The advantages of the "full PKI" case derive from the long experience with PKI technology, deployment, and services,
477 the substantial number of PKI TTPs and enterprise CAs, and the best practice qualities of PKI for key management in
478 large populations. For these reasons, modern high- and medium-assurance trust management infrastructures tend to
479 be constructed around PKI.

480 **5.2.2. Example: Delegated Trust Scenario**

481 The general Liberty architecture model is that a principal authenticates to a service provider via an identity provider.
482 This example identity provider model describes a case where the identity provider function is distributed and collocated
483 with individual principals. For this case, new trust aspects must be taken into account because this model introduces a
484 new element in the trust chain. Indirect trust is applied through certification, to enable individual identity providers
485 to be validated by the entities accepting their assertions.



486

487

Figure 5. Delegated identity provider Model

488 In this model, the service provider does not have a direct agreement with the principal's identity provider, but trusts
489 the issuer (acting as a CA) to establish indirect trust. The issuer uses its key to certify the principal's identity provider,
490 thereby establishing a chain that can be verified by any entity obtaining the issuer's public key. Typically, certification
491 of principals' identity providers by issuers would take place as part of the registration process between the principal
492 and the infrastructure that the issuer represents. A service provider can trust a principal based on the certificate that
493 his/her identity provider presents, when the service provider has a (direct or indirect) trust relationship with the issuer.
494 Note that several issuers may certify a single principal's identity provider.

495 The principal's identity provider must store the private key corresponding to its certificate in a secure way, because it
496 is essential to guarantee that no one can masquerade as the principal. In practice, this will require the usage of smart
497 cards or other tamper resistant media to securely support the distributed identity provider case.

498 One practical example of this kind of model is a mobile Liberty client, where the identity provider provides its
499 certificate to the mobile terminal and the service provider trusts the issuer. Based on this trust, the service provider
500 can also trust the certificate stored in the mobile client.

501 **6. Comparison Among Models**

502 As the preceding sections demonstrate, a variety of methods can be employed to establish trust among Liberty
503 processing components, achieving different types and levels of assurance. Cryptographic authentication may be
504 based on direct exchange of keys between peers or may be indirect through one or more intermediaries, and may
505 employ a variety of public-key and secret-key technologies. Similarly, the business agreements enabling transactions
506 may be directly exchanged between peers, may be indirect through one or more intermediaries, may be absent
507 or unnecessary for particular transactions, and/or may be derived from enrollment and participation in a shared
508 authentication infrastructure. Authentication trust and business trust may vary independently, thereby supporting
509 a broad range of operational environments.

510 Liberty Phase 1 presumes direct business agreements among the set of entities comprising a circle of trust, employing
511 the Pairwise Trust model. It requires certificate-based authentication of identity providers, and recommends its use
512 for other purposes (authentication of service providers, signing of assertions), but is silent as to whether the trust
513 model applied to verify those certificates is direct or indirect. Pairwise Trust enables strong bonds of mutual trust to
514 be developed, but impedes connectivity beyond small, closed communities. Brokered Trust and Community Trust
515 represent two alternative strategies to enable broader sets of entities to interoperate with one another.

516 Liberty Phase 2 introduces the prospect that identity providers may operate as intermediaries, introducing service
517 providers with which they share relationships to other identity providers; this comprises the Brokered Trust model.
518 Relative to Pairwise or Community Trust, it adds complexity by interposing active, trusted entities into the protocol
519 transactions performed to accomplish federation. On the positive side, it centralizes the management of interdomain
520 relationships at a relatively small number of entities.

521 Cryptographic trust establishment infrastructures can be used to enable broader secure interoperability than would be
522 practical if direct authentication trust needed to be established among pairs of participants; this approach exemplifies
523 the Community Trust model. Relative to Brokered Trust, it simplifies federation transactions, at the cost of
524 making larger numbers of entities responsible for assessing and managing cross-domain relationships. Where
525 business requirements permit, use of Community Trust can obviate the need to deploy and invoke the intermediary
526 identity providers that are characteristic of Brokered Trust. If independent organizations interested in facilitating
527 communications among entities (e.g., a community or an industry consortium) were to deploy or sponsor infrastructure
528 facilities, such resources could help to facilitate and encourage the growth of Liberty-based connectivity.

529 For Liberty to achieve its potential benefits, interoperability beyond the scope of small, closed communities must be
530 possible. Deployers should recognize the prospects of the Brokered and Community Trust models, and should select
531 the choice that best fits their business and operational requirements.

532 7. Trust Establishment Mechanisms

533 This chapter introduces an overview and essential characteristics about trust establishment mechanisms applicable to
534 Liberty. PKI and Kerberos can be seen as the primary candidate methods for this purpose.

535 7.1. Public Key Infrastructure, PKI

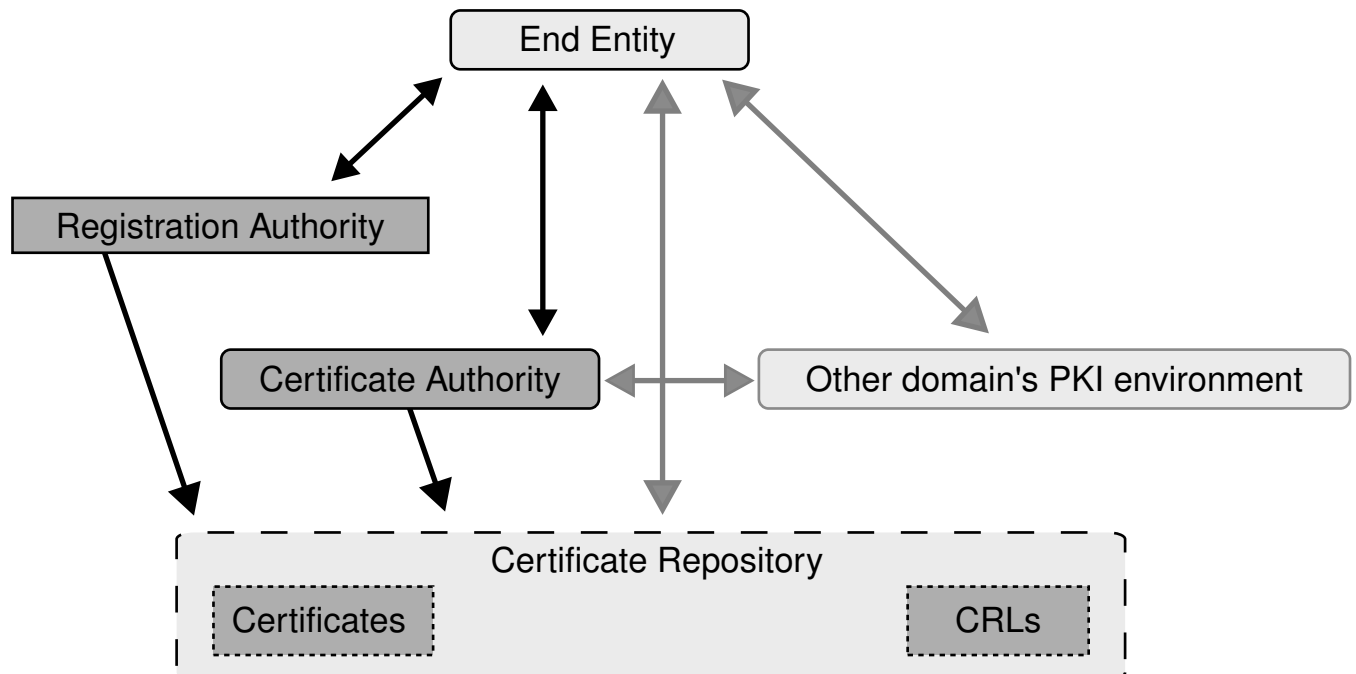
536 PKI-based approaches can provide secure, trusted and efficient key and certificate lifecycle management. This
537 facilitates security services like authentication, data integrity, confidentiality and non-repudiation, which are often
538 seen as essential components and building blocks of modern security. Discussions of PKI and its deployment and
539 usage are available in numerous publications, e.g. [Adam99] [Hous01] [Nash01].

540 In PKI, a certificate serves to bind a named entity to a public key. The most common and standardized certificate format
541 is ITU-T X.509 (currently version 3) [X.509], discussed in Section 7.1.1. PKI system deployments using standard
542 X.509v3 certificates include the following main components (not all of which are required in all configurations):

- 543 • Public-key certificate;
- 544 • Certification Authority (CA);
- 545 • Registration Authority (RA);
- 546 • Certificate Repository;
- 547 • End entity (user).

548 This section introduces these building blocks and Section 7.1.2 outlines the various trust models currently in use.

549



550

551

Figure 6. PKI Elements

552 **End-Entity (EE)** a user of PKI certificates and/or end-user system that is the subject of a certificate.

- 553 **Certificate Authority (CA)** Acts as the signer of certificates. Primary tasks include the issuance of certificate,
554 renewal of certificate and revocation of certificate.
- 555 **Certificate Repository (CR)** Stores the issued certificates and Certificate Revocation Lists (CRL). Usually pro-
556 vides an interface for users to search directory (such as LDAP interface or HTTP)
- 557 **Registration Authority (RA)** Optional element in PKI system and can be combined with the CA. RA can do
558 some of the CA's management functions and can therefore take some of the load off
559 from CA. RA registers users into the PKI infrastructure. It is particularly useful to
560 separate the RA component when the CA is remote and the RA registers the users in
561 person on behalf of the CA.

562 7.1.1. X.509

563 X.509 is the common name by which the International Standard defining the PKI Framework is known. It is also the
564 term that is generally used to identify public-key certificates formatted in accordance with the standard. The X.509
565 standard has been updated and enhanced several times. Some of the revised editions of the standard enhanced the
566 fundamental structure of a public-key certificate and therefore resulted in a new "version" of public-key certificates.
567 The 1st edition of the X.509 standard was first published in 1988 and the certificates defined in that edition were known
568 as X.509 v1 certificates. The 2nd edition of the X.509 standard was published in 1993. It enhanced the certificate
569 structure, resulting in X.509v2 certificates, by adding two new elements (issuerUniqueID and subjectUniqueID). The
570 3rd edition of the X.509 standard was published in 1997 and resulted in the definition of the X.509 v3 certificate format.
571 V3 certificates extended the v2 format by adding a general extensions mechanism. As a result of this mechanism, no
572 further certificate versions are anticipated. A number of certificate extensions were defined in the 3rd edition. The
573 4th edition of X.509 was published in 2000 and although it defined an additional set of certificate extensions, no new
574 certificate format was required. Certificates that include these new extensions are X.509 v3 certificates. The X.509
575 v3 specification is profiled for Internet usage in IETF [[RFC3280](#)].

576 The main purpose of an X.509 certificate is to establish a link between an identified entity and a public key (and,
577 indirectly, with the corresponding private key held confidentially by the entity). This is accomplished by signing the
578 certificate using the private key of a CA, so that the certificate can subsequently be verified by any entity holding or
579 obtaining the CA's public key. The public keys carried in certificates can be used for signature or encryption purposes.
580 When signatures are required, a principal applies a private key and relying parties verify that signature using the public
581 key in the entity's certificate. To perform encryption, the public key in a subject's certificate is used and the subject
582 may decrypt the data using their corresponding private key. Commonly, public-key encryption is used to transfer a
583 symmetric key, which is used in turn for encryption of message data. Typically, users will have two public key pairs
584 (and two corresponding certificates), one for digital signature purposes and a separate set for encryption purposes.

585 When a certificate is issued, it asserts a binding between a named subject and a public key for a predetermined
586 validity period. When a certificate is used, it is important to determine that its contents remain valid. Two classes
587 of approaches have been specified for this purpose: Certificate Revocation Lists (CRLs, defined within the X.509
588 specification) and on-line certificate status checking services (e.g., Online Certificate Status Protocol [[RFC2560](#)]
589 and XML Key Management Specification [[XKMS](#)]). Generally, on-line services can offer more timely detection of
590 revocation events, but require access to trusted and available responders; CRLs are best suited to providing revocation
591 information on a scheduled basis.

592 Information included in a X.509v3 certificate:

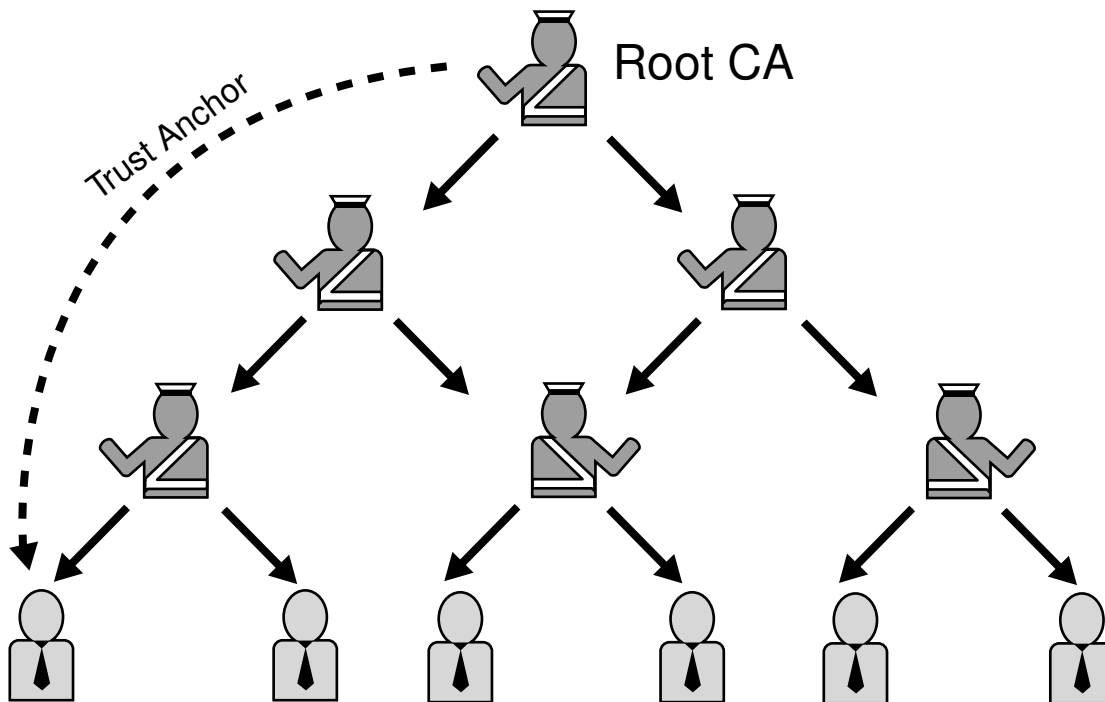
- 593 1. Public key of certificate owner;
- 594 2. Issuer's (CA) individual name;
- 595 3. Validity time of certificate;

- 596 4. Subject, name of the certificate owner;
- 597 5. Digital signature of the issuer;
- 598 6. Extensions.

599 7.1.2. Trust establishment in PKI system

600 PKI enables a variety of different trust models. The selection of a trust model for a certain environment depends on
601 several different factors and the requirements for one environment can vary greatly from those in another. Trust models
602 for Liberty were introduced in previous chapters of this document.

603 The three primary trust models used in PKI are hierarchical, distributed and bridge. Hierarchical trust is a common PKI
604 trust model. In this model the trust is established as a tree structure from top to bottom. At the top of the whole trust
605 model is the root CA that has sub-CAs, with sub-CAs providing CA services to their end entities. In the hierarchical
606 trust model, there is a single trust anchor, the public key of the root CA that is used by all relying parties within the
607 hierarchy.



608
609

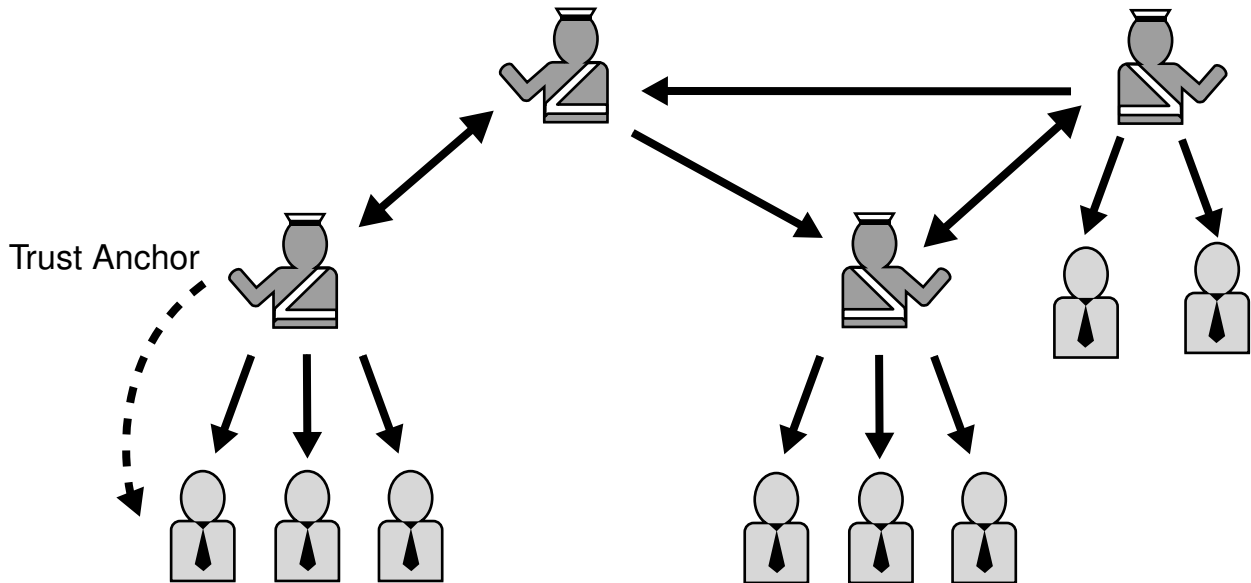
Figure 7. Hierarchical PKI Model

610 This kind of trust model makes possible to delegate trust and CA operations to sub authorities. When the trust chain is
611 built in this model, it is done by backtracking. The path must be built from the end entity up to the root CA. Once the
612 path is built, however, processing of the certificates must be done in order from the trust anchor down to the end-entity
613 certificate.

614 The hierarchical trust model is best suited to environments where there is a natural root identified for the business
615 environment and there is a fully established development process for the architecture in place.

616 *The distributed trust model* is one where no single CA roots all trust. Rather, typically the key used as a trust anchor
617 for a given user is the public key of the CA that issues certificates to that user. In this model, there is a distributed
618 network of trust anchors. One advantage of this model is that there is no single point of failure as there is with the

619 single trust anchor in the hierarchical model. Also, in this model the CAs are able to act fairly autonomously without
620 being bound by policy delegated from a root CA.



621

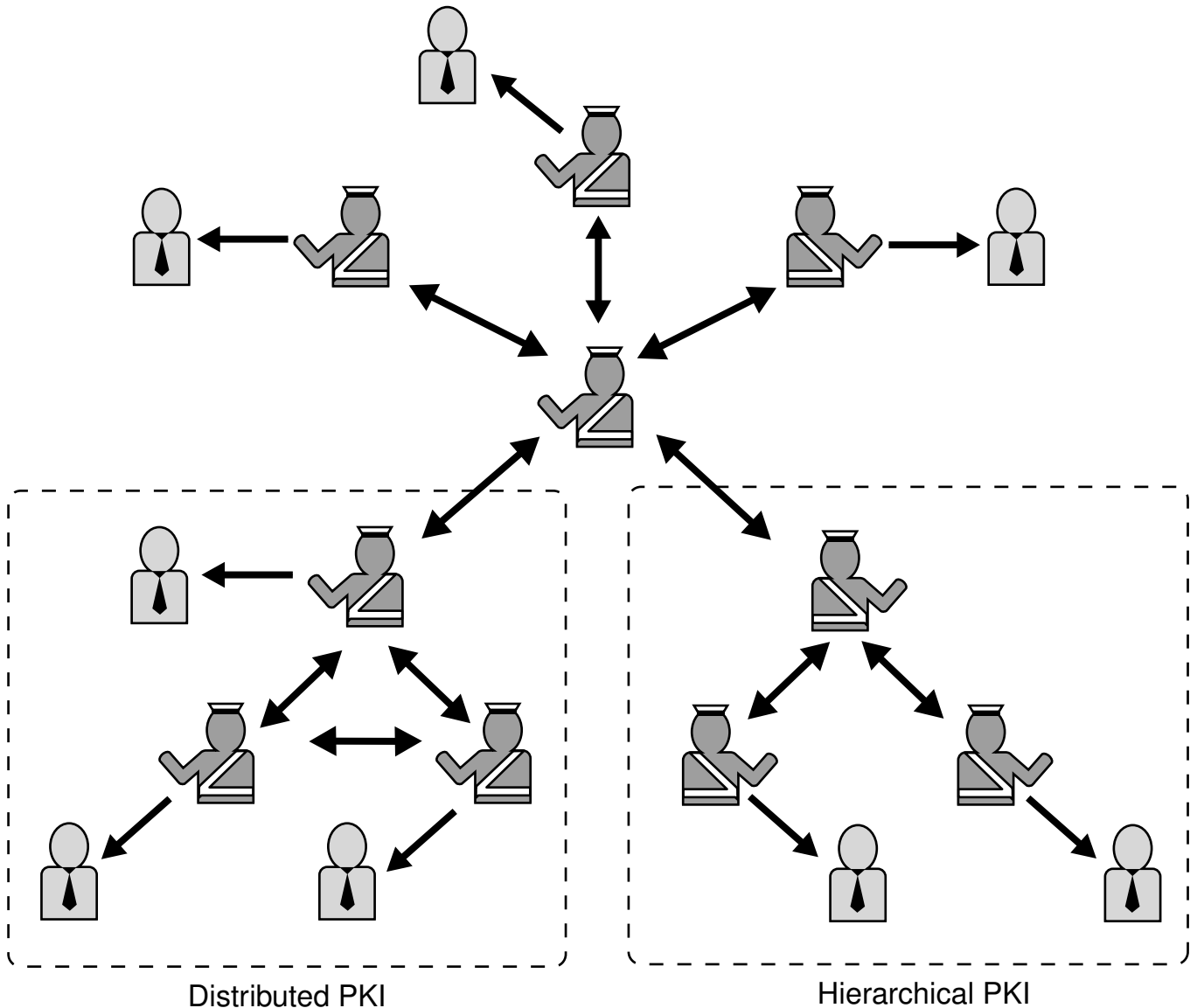
622

Figure 8. Distributed PKI Model

623 In the distributed trust model, certification paths can be built in either direction, or a combination of both, however,
624 processing of the certificates must always be done from the local trust anchor to the end-entity certificate. The
625 distributed trust model is best suited to business-to-business environments where there are a relatively small number
626 of CAs that need to be inter-connected.

627 *The Bridge trust model* is similar to the distributed model in that there is no single Root CA and no single trust anchor
628 common to all users. In the bridge model there is a single CA that acts purely as a facilitator to interconnect other
629 CAs. A bridge CA typically does not issue certificates to any end entities, but is used, as a hub, to interconnect the
630 spokes which can be individual CAs, PKIs that use the hierarchical trust model, and PKIs that use the distributed trust
631 model. The primary benefit provided by a bridge CA is that each spoke need only maintain a single cross-certification
632 with the bridge CA and they are automatically able to build certification paths across all spokes in the model.

633



634

635

Figure 9. Bridge PKI Model

636 In the bridge model, certification paths can be built in a combination of directions. If the path includes certificates in
637 a hierarchical PKI, those portions of the path would be built from the end-entity to the root of that hierarchy. Other
638 portions of the path can be built in either direction. Processing of the certificates in the path, as with the other trust
639 models, must always be done from the trust anchor to the end-entity certificate. The bridge trust model is best suited
640 to environments where a large mesh of cross-certificates would otherwise be needed to establish the required trusted
641 environment, such as the U.S. Federal Government and its agencies. The bridge CA can also provide a single point of
642 interconnection for all its spokes to external PKIs.

643 In all models described above, trust between CAs is established by using *cross-certification*. In the hierarchical model,
644 cross-certification is used to delegate responsibility to subordinate CAs. It is also used for connecting the hierarchical
645 PKI to other certification domains. In the distributed model cross-certification is used to connect the CAs within
646 a domain and similarly to connect the spoke CAs with the hub, the bridge CA. Cross-certification can be seen as
647 representing a peer-to-peer contract between two CAs.

648 In cross-certification trust establishment, CAs create trust to each other so that CA A's entities are trusted by CA B's
649 entities and issue cross-certificates to represent these trust relationships. Cross certificates can include extensions that
650 impose constraints on the set of certificates in the remote domain that are acceptable to be trusted by relying parties
651 in the local domain. Depending on applicable policies, cross certificates may be issued by root CAs or by sub-CAs
652 within their hierarchies.

653 Additional discussion on PKI trust models can be found, e.g., in [Elle01], [Linn00], and [Per199].

654 **7.1.3. Conclusions**

655 Public key cryptography enables strong methods for entity authentication and PKI provides many methods to establish
656 trust relations between different entities. The appropriate architecture for each situation can be determined based on
657 numbers of entities, numbers of CAs, and their organizational relationships and associated policies.

658 **7.2. Kerberos**

659 Kerberos [RFC1510] is the most common method to provide strong authentication between users and servers by
660 using secret key cryptography, based on a protocol developed by MIT. After the identity is proved both entities can
661 communicate using encryption and integrity protection.

662 Kerberos provides key freshness, i.e., a new session key is created whenever two entities want to communicate with
663 each other. Since new keys are generated for each session, an attacker that determines the key used for one session
664 cannot use it to decrypt subsequent traffic.

665 In Kerberos, each participating user and server shares a distinct long-term secret key with a trusted authority, the Key
666 Distribution Center (KDC). For the user case, the shared secret is derived from a password. These secrets are used
667 for processing at their respective entities, but are not transmitted over the network. Session keys are generated and
668 delivered by the KDC within protocol elements called tickets, when communication between two entities is starting.
669 In most current Kerberos deployments, the key shared between entity and KDC is a (56-bit) DES key, with DES CBC
670 mode used for encryption. Specification activities incorporating triple-DES (112-bit key) and AES (128-bit and longer
671 keys) are currently in progress, and use of these newer algorithms appears prudent from a cryptographic perspective
672 once corresponding implementations are available.

673 Once registered with a KDC, a user's Kerberos interactions proceed as follows. The user's client requests a special
674 type of ticket (the Ticket Granting Ticket, or TGT) from the KDC, receives the TGT and an encrypted representation
675 of the corresponding TGT session key, and applies the user's password to decrypt the TGT session key. Once this
676 step is complete, the user's password can be deleted from memory, as it is not required for subsequent use of the
677 TGT. When the user wishes to communicate with a particular server, it sends the KDC a message with its TGT, an
678 authenticator based on knowledge of the TGT session key, and an indication of the server with which the user wishes
679 to communicate. If the KDC successfully validates the authenticator, it generates a service ticket for the user to use
680 in communication with the requested server and returns it to the user's client along with a representation of the service
681 ticket's session key, encrypted using the TGT session key. Based on this data, an authentic client can now generate
682 an authenticator with the service ticket session key and can send it to the server along with the service ticket, thereby
683 authenticating its user to the server.

684 **7.2.1. Kerberos Processing**

685 This section describes the basic Kerberos cryptographic protocol based on Kerberos version 5. As preconditions, both
686 the user and server have keys that are registered with the KDC. The user's key is generated from the password he/she
687 has chosen, and the server's key is randomly selected and stored at the server.

688 Processing in Kerberos:

- 689 • User A sends a message to KDC and tells the KDC that it wants to communicate with server B

- 690 • KDC creates a random session key, K and makes two copies of it. KDC creates two encrypted messages, where
691 message 1 (m1) is encrypted with user's key and message 2 ("ticket") with server's key. Both messages are sent to
692 user.
693 $m_1 = e_{K_A}(K, ID(B))$
694 $m_2 = e_{K_B}(K, ID(A))$
- 695 • User decrypts the message 1 with his/her own key and gets the session key.
- 696 • User creates new message ("authenticator"), m3, and encrypts it with new session key. This new message
697 includes the timestamp T. Timestamp is included to prevent the sending the message 2 again later by attacker
698 who impersonates the user.
699 $m_3 = e_K(ID(A), T)$
- 700 • User sends messages 2 and 3 to the server
- 701 • Server decrypts message 2 with the key it shares with KDC and gets the session key. Then it decrypts the message
702 3 with new session key.
- 703 • If the user wants the server to be authenticated as well, an additional message is needed. In this case the server
704 takes the timestamp, T, from the message 3 and creates new message, m4, which is encrypted with session key.
705 $m_4 = (ID(B), T)$

706 7.2.2. Conclusion

707 The basic Kerberos protocol is vulnerable to password guessing attacks against TGTs, as a TGT can be requested
708 and obtained without first demonstrating possession of the password; the optional preauthentication facility provides
709 a countermeasure against this attack. Interoperability between different realms can be accomplished using inter-realm
710 protocol facilities and shared inter-KDC keys, but trust models for inter-realm Kerberos operation have received less
711 evaluation and standardization than corresponding models for PKI environments.

712 **8. Integrating Trust Establishment Infrastructures with Liberty**

713 In practice, trust establishment technologies would be applied in a layered fashion to support Liberty requirements.
714 At the lowest level, a bootstrapping process would be used to create and maintain authentication trust among the
715 participating entities: an entity would initially be enrolled in a trusted relationship with a trust establishment service.
716 The trust establishment service would then facilitate introductions between this and other enrolled entities.

717 The nature of enrollment with the trust establishment service and the mechanisms for authentication trust between a
718 Liberty entity and the trust establishment service are unspecified by Liberty. Authentication trust could be achieved
719 by any technical mechanism that provides message authenticity, integrity, and confidentiality, e.g., physically secure
720 channels, PKI, manual SKI, or Kerberos for authentication, together with SSL/TLS, IPSEC, S/MIME, or SSH for
721 integrity and confidentiality.

722 The nature of authentication trust between Liberty entities, as delivered by the trust establishment service, is partially
723 defined in the Liberty specifications. Some entities are required to accept SSL/TLS sessions, and all are required
724 to verify XML-Signatures [RFC3275] on messages if present. Although the Phase 1 Liberty specifications do
725 not require all XML messages to be signed, it is best practice for senders to sign all messages, and the Phase 1
726 Liberty specifications note that vulnerabilities may be introduced if messages are not signed. Some entities may
727 initiate SSL/TLS sessions with certificate-based authentication. Liberty entities may use additional mechanisms
728 that are permitted, but not required, in the Liberty specs, for example, IPSEC security associations. All of these
729 security mechanisms require the distribution of cryptographic keys (public/private key material and/or symmetric key
730 material). The primary function of the trust establishment service is the distribution and management of this key
731 material. Once private or symmetric keys are distributed, secure processing depends on protection of the stored keys
732 against compromise; while such protection mechanisms are implementation-specific and are not defined by Liberty
733 specifications, they are important aspects of secure processing components.

734 The Liberty Phase 1 specifications do not mandate XML-Signatures on all messages, nor do they constrain the
735 technical options present in XML-Signature when it is used. These options include, for example, signature systems
736 based on both PKI (using asymmetric key pairs) and HMAC algorithms (using symmetric keys). Implementation
737 requirements may favor one or another approach, however, because of the advantages of PKI for key distribution and
738 non-repudiation, best practice for large-scale deployments will generally use PKI mechanisms. PKI may imply,
739 however, some additional system complexity and costs. Small-scale systems, or systems that create no questions of
740 legal liability (e.g., a Liberty deployment entirely within a single company), might rely on secured channels between
741 Liberty elements, or manual, symmetric keying for signatures.

742 Since X.509v3 certificates can be used to implement authentication trust in the SSL/TLS and XML-Signature protocols
743 named in the Liberty Phase 1 specifications, the trust establishment service may, in fact, be an X.509v3 Certification
744 Authority, providing usual and customary CA services. Advantages of this approach are the significant number of
745 commercial Trusted Third Parties (TTPs) already providing these services, the large number of compatible software
746 implementations available, and the broad dissemination of technical knowledge concerning PKI. TTP services include
747 the ability to certify participating Liberty providers, distribute issued certificates, and update and distribute Certificate
748 Revocation Lists. Additionally, on-line validation services (e.g., through the OCSP or XKMS protocols) could be
749 provided for the certificates. This model offers scalable trust at a strong level (though somewhat less than that of the
750 Circle of Trust Model), but requires organizational involvement to establish and manage infrastructure.

751 The term "trust establishment service" is used in a general sense, because although the service could, in fact, be a
752 Certification Authority, the service need not operate as a conventional CA. Instead, it could be a broker for several
753 CAs (e.g., a PKI bridge). It could deliver private keys and certificates through protocols not conventionally associated
754 with CAs (e.g., in files through a shared file system). It could construct certificates in unconventional ways (e.g.,
755 all participating entities use the same private key and the same short-lived certificate, replaced daily). Researchers,
756 companies, and governments continually seek improvements to the technology of trust management, and many new
757 alternatives will appear and be tested by the marketplace.

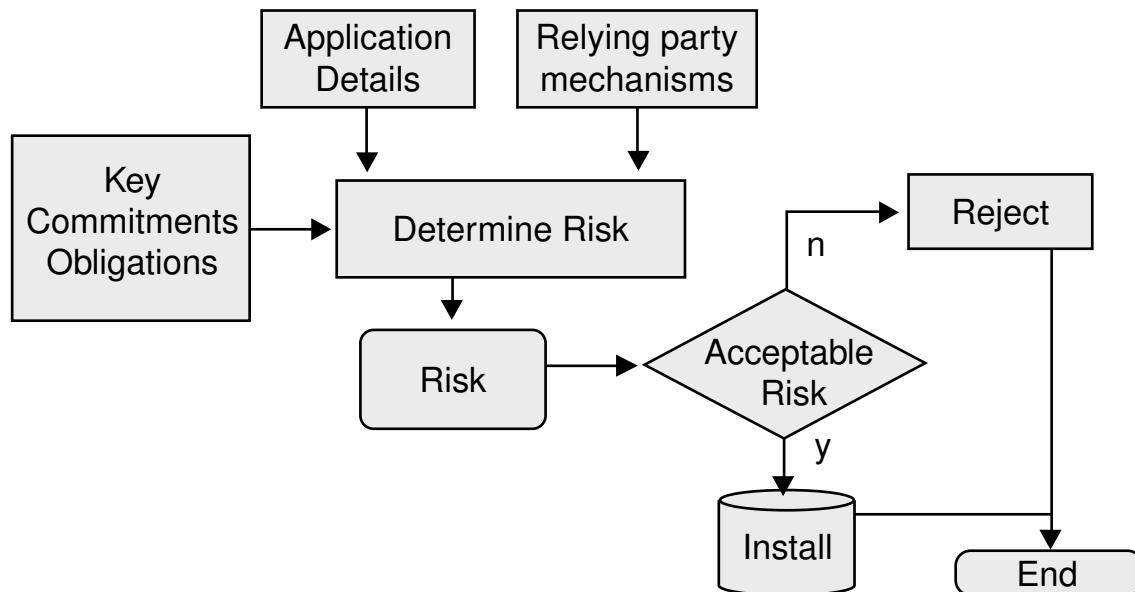
758 9. Metadata and Trust Discovery

759 If two entities attempt to communicate without previous awareness of membership in a common trust infrastructure,
760 the following outcomes are possible:

- 761 1. the entities communicate insecurely without authentication
- 762 2. the entities transfer data enabling them to perform authentication
- 763 3. the entities do not interoperate

764 In the second scenario, an entity wishes to communicate with another entity in order to perform some transaction but
765 has no pre-existing basis for the required technical trust. Nevertheless, the entities may be able to establish trust
766 between themselves through exchange of trust metadata.

767 One such mechanism would be for the involved entities to publish their public keys along with their approved usages,
768 the commitments the key owner makes with respect to that key, and the obligations that a relying party must accept
769 (either implicitly or explicitly) if were to use that key. The key owner would publish this statement to potential
770 relying parties; an XML Signature calculated over it would both ensure its integrity and bind the associated private
771 key to those statements. A relying-party, once it discovered this signed statement, would be able to examine the
772 approved applications, commitments and obligations associated with that key and determine whether or not the key
773 was appropriate to an intended application. If the result of this analysis were positive, the relying party would install
774 the public key into some trusted store - the stored key indexed by the application usages for which it was appropriate.
775 This process is shown in the following diagram.



776

777

Figure 10. Validation of Key from Metadata

778 As the public key is distributed along with the associated business commitments and obligations, exchange of Trust
779 Metadata in this scenario can be thought of enabling both business and authentication trust (e.g. a decision to install a
780 key will result in the addition of the key-owner to both of the relying-party's BAL and TAL).

781 References

782 Normative

- 783 [RFC1510] Kohl, J., Neuman, , C., eds. (September 1993). "The Kerberos Network Authentication Service (V5),"
784 RFC 1510, Internet Engineering Task Force <http://www.rfc-editor.org/rfc/rfc1510.txt>
- 785 [RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C., eds. (June 1999). "X.509 Public Key
786 Infrastructure: Online Certificate Status Protocol - OCSP," RFC 2560, The Internet Engineering Task Force
787 <http://www.rfc-editor.org/rfc/rfc2560.txt>
- 788 [RFC3275] Eastlake , D., Reagle, J., Solo, D., eds. (March 2002). "(Extensible Markup Language) XML-
789 Signature Syntax and Processing," RFC 3275, The Internet Engineering Task Force [http://www.rfc-
editor.org/rfc/rfc3270.txt](http://www.rfc-
790 editor.org/rfc/rfc3270.txt)
- 791 [RFC3280] Housley, R., eds. (April 2002). "Internet X.509 Public Key Infrastructure Certificate and Certifi-
792 cate Revocation List (CRL) Profile," RFC 3280, The Internet Engineering Task Force [http://www.rfc-
editor.org/rfc/rfc3280.txt](http://www.rfc-
793 editor.org/rfc/rfc3280.txt)
- 794 [Elle01] Elley, Y., Anderson, A., Hanna, S., Mullan, S., Perlman, R., Proctor, S., eds. (2001). "Building Certification
795 Paths: Forward vs. Reverse," ISOC NDSS
- 796 [Linn00] Linn, J., eds. (6 November 2000). "Trust Models and Management in Public-Key Infrastructures," RSA
797 Laboratories Technical Note <http://www.rsasecurity.com/rsalabs/technotes/index.html>
- 798 [XKMS] Hallam-Baker, P., eds. (31 January 2002). "XML Key Management Specification (XKMS 2.0)," Working
799 Draft, W3C <http://www.w3.org/2001/XKMS>
- 800 [X.509] "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate
801 frameworks," ITU-T (2000). ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-8:2000,

802 Informative

- 803 [Adam99] Adams, C., Lloyd, S., eds. (1999). "Understanding the Public-Key Infrastructure: Concepts, Standards,
804 and Deployment Considerations," New Riders Publishing
- 805 [Hous01] Housley, R., Polk, T., eds. (2001). "Planning for PKI: Best Practices Guide for Deploying Public Key
806 Infrastructure," John Wiley and Sons, New York
- 807 [Nash01] Nash, A., Duane, W., Joseph, C., Brink, D., eds. (2001). "PKI: Implementing and Managing E-Security,"
808 Osborne/McGraw-Hill, New York
- 809 [Per199] Perlman, R., eds. (November/December 1999). "Overview of PKI Trust Models," IEEE Network