分散システム

第5回 名前

双見 京介（FUTAMI Kyosuke）
高田秀志（TAKADA Hideyuki）
2025年11月

はじめに

- 分散システムにおいて，資源の共有，エンティティの同定，場所の参照を行うために，**「名前」**が用いられる
- プロセスが名前付けされたエンティティにアクセスできるように，**「名前付けシステム」**は**名前解決**を行う
  - フラットな名前付け
  - 構造化された名前付け
  - 属性ベース名前付け
- 分散システムにおける名前付けシステムは，それ自体が複数のコンピュータにまたがる分散システムで実現されている

名前付け

- 「名前」は，エンティティ（プリンタ，ユーザ，ファイルなど）を参照するための文字列あるいはビット列
  - 名前（name）
  - アドレス（address）
  - 識別子（identifier）
- 「アドレス」は，エンティティのアクセスポイントの名前
- 「識別子」は，以下の性質を持つ名前
  - 1つの識別子は複数のエンティティを参照しない
  - 各エンティティは少なくとも1つの識別子により参照される
  - ある識別子は常に同じエンティティを参照し，再利用されない

名前，アドレス，識別子

- アドレスや識別子は，ビット列のような機械読み込み可能な形式（**machine-readable form**）でのみ表現
- **ヒューマンフレンドリーな名前**（**human-friendly name**）は，人間にとって読みやすいような形式を持つ
  例　/Users/htakada/Documents/slides.ppt
- 名前や識別子はアドレスに変換される必要がある（**name-to-address binding**).

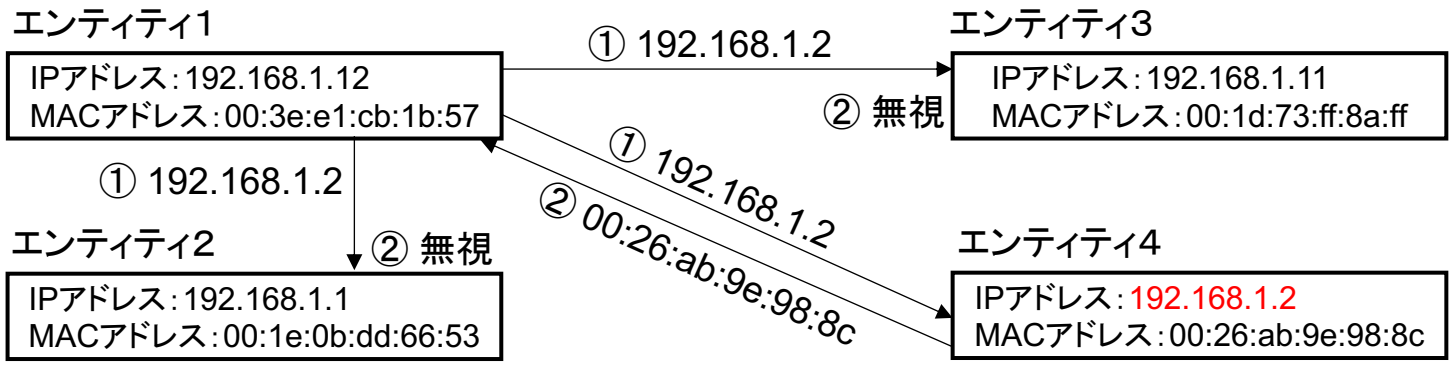
  例　www.ritsumei.ac.jp → 133.19.170.14
  ヒューマンフレンドリーな名前　　IPアドレス

フラットな名前付け

- 名前に構造を持たない（フラットな）場合に対する名前解決
- エンティティの識別子が与えられたときに，そのエンティティへのアクセスポイント（アドレス）を得る
  - ブロードキャスト方式
  - 転送ポインタ（Forwarding Pointers）
  - ホームベースアプローチ（Home-Based Approach）
  - 分散ハッシュテーブル（Distributed Hash Tables）
  - 階層的アプローチ（Hierarchical Approach）

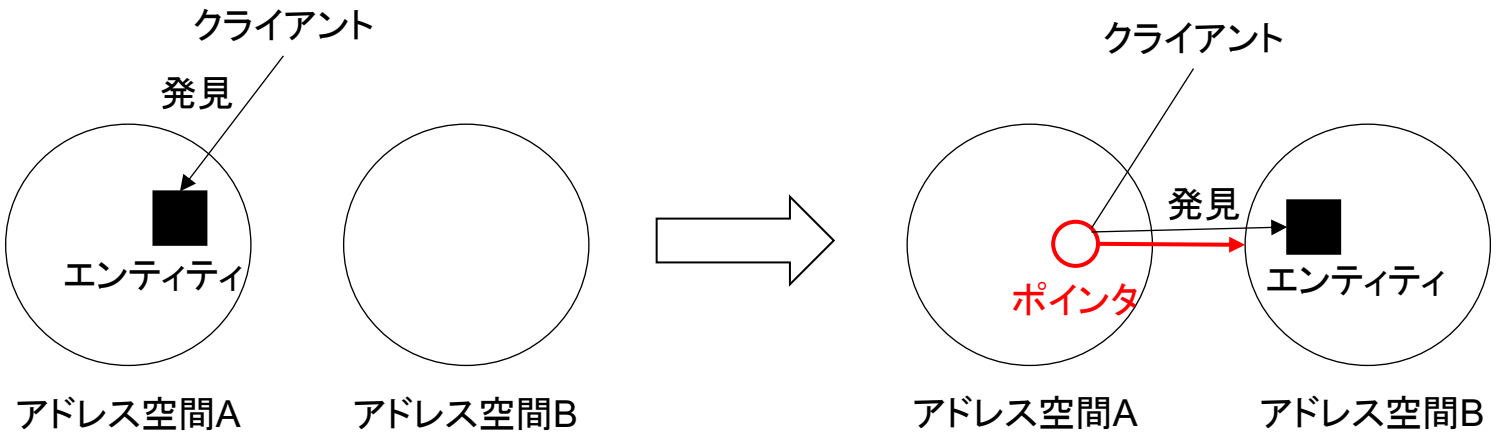
ブロードキャスト方式

- Address Resolution Protocol（ARP）で利用
- IPアドレスを元に，通信に必要なMACアドレスを得る
- （例）エンティティ1がエンティティ4と通信したい場合
  - ① エンティティ1がすべてのエンティティに通信先のIPアドレスを自身のIPアドレス・MACアドレスと共に送信
  - ② 受信したIPアドレスが自分のものと一致した場合は，自身のMACアドレスをエンティティ1に送信，一致しない場合は無視



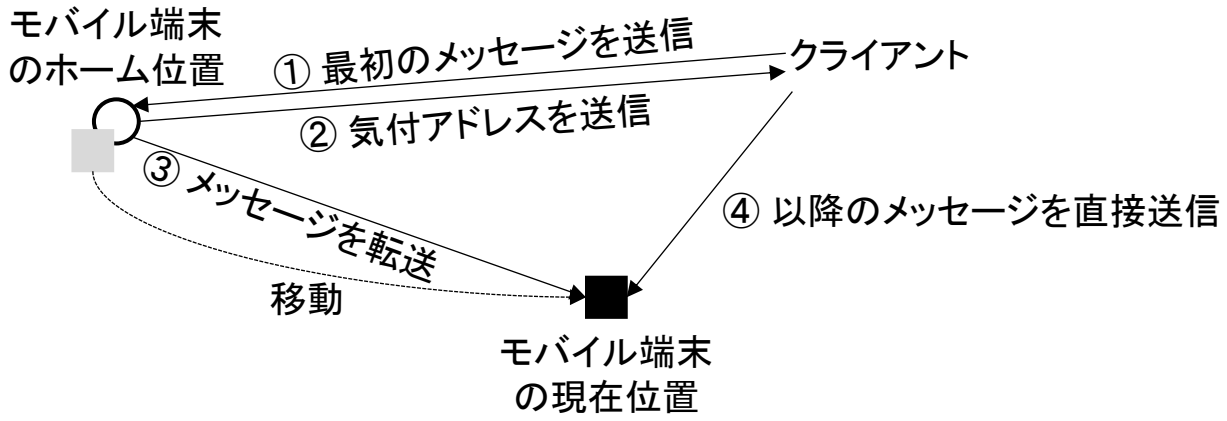
転送ポインタ

- エンティティがアドレス空間Aからアドレス空間Bへ移動するとき，アドレス空間Aにアドレス空間Bへのポインタを残す
- クライアントは，エンティティが移動したとしても，ポインタをたどることによって移動先のエンティティを発見可能
- ポインタの連鎖が長くなると効率が悪くなる。また，連鎖のうち一つでも失われると，移動先エンティティにたどり着けない

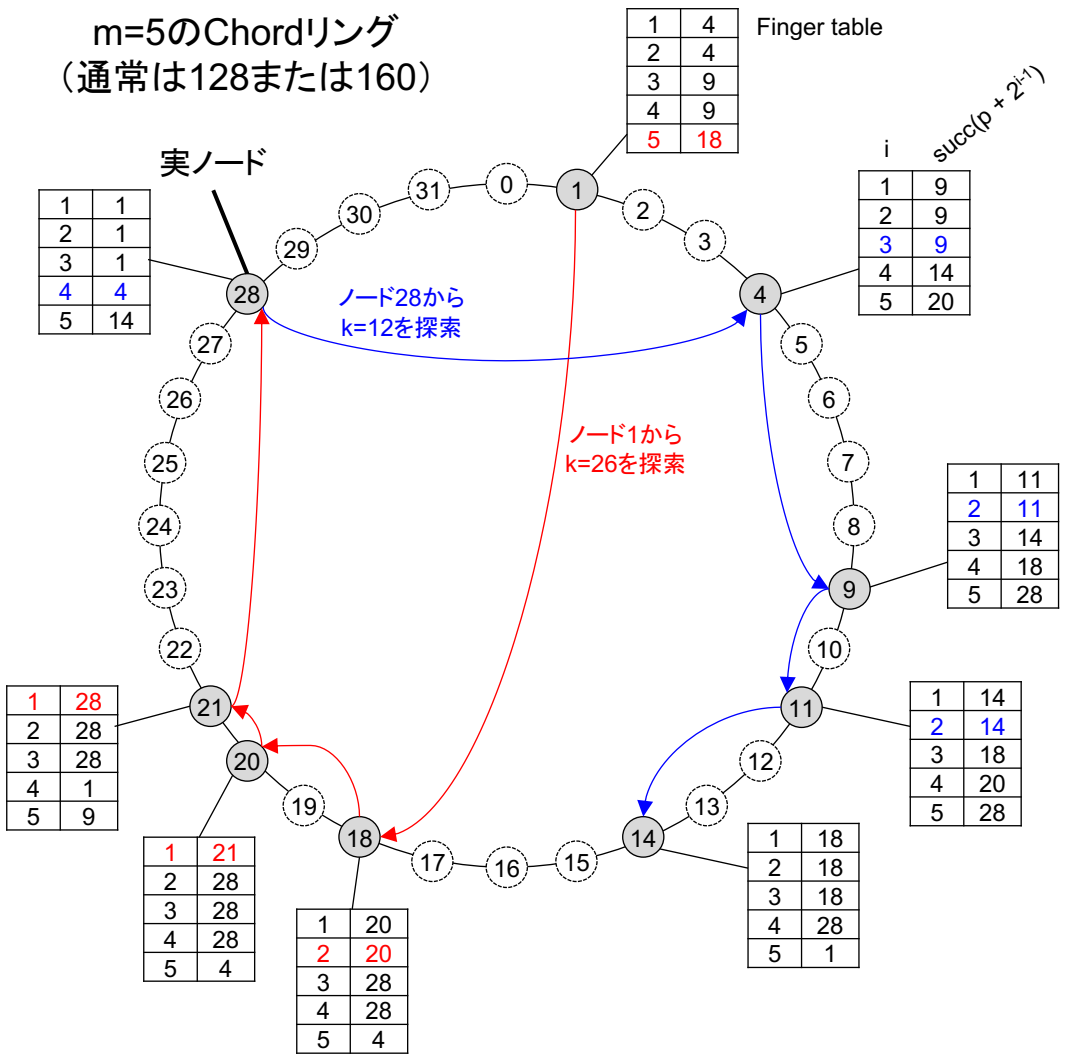


ホームベースアプローチ

- 大規模なネットワーク内でのモバイルエンティティの管理
- モバイルIPで利用
  - モバイル端末のホーム位置として固定のIPアドレスを割り当て（ホームエージェント）
  - モバイル端末はホームエージェントに自身の現在のアドレス（気付アドレス）を登録
  - クライアントからモバイル端末に対する最初のメッセージはホームエージェントに対して送信（①）
  - ホームエージェントは登録されているモバイル端末の気付アドレスをクライアントに送信（②）し，メッセージをモバイル端末へ転送（③）
  - 以降，クライアントはモバイル端末の気付アドレスを用いて通信（④）

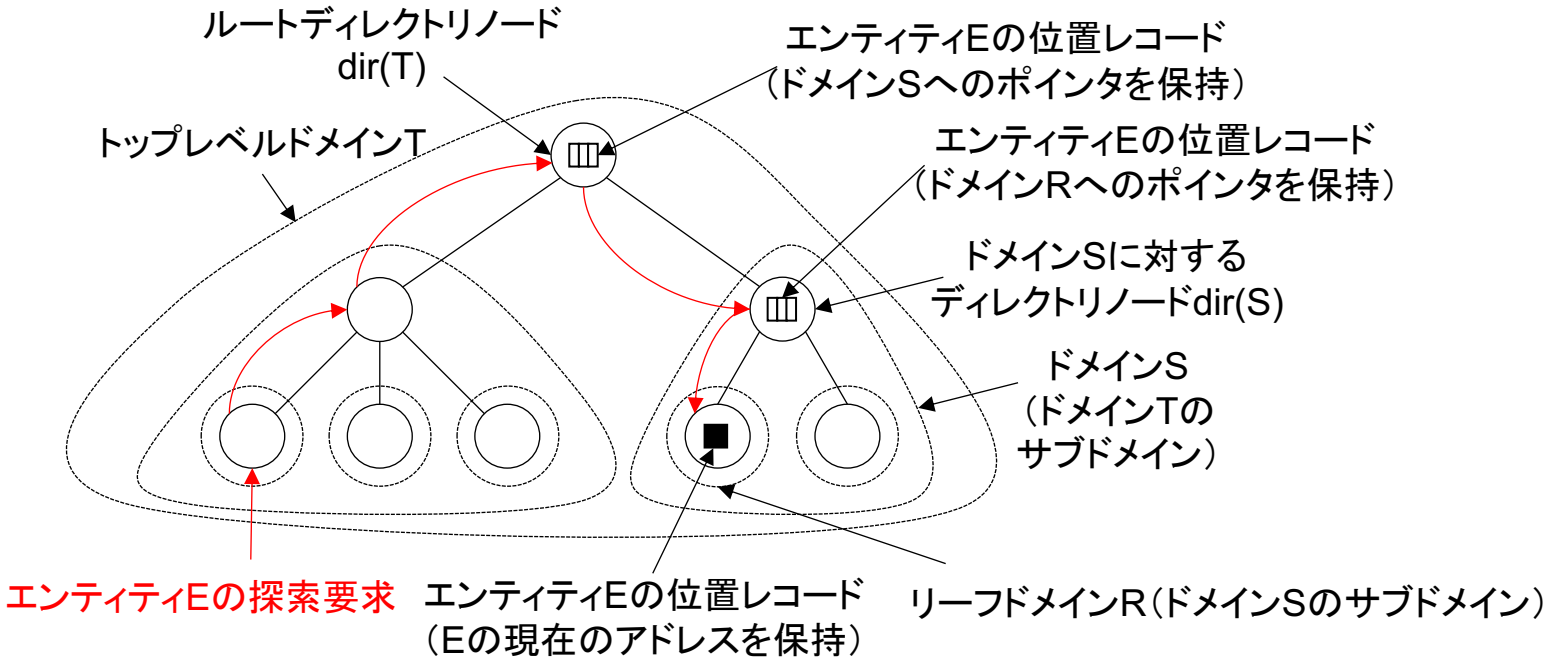


分散ハッシュテーブル（Chord）



- mビットの識別子でリングを構成
- 2^m個の識別子の中からランダムに実ノードを割り当て
- キーkを持つエンティティの情報はid≧kとなる最小の識別子idを持つ実ノード（succ(k)）に格納
- 効率化のため，実ノードはiと$succ(p + 2^{i-1})$のペアを格納したFinger tableを保持
- キーkが与えられたとき，succ(k)を求めたい
  - いずれかの実ノードから探索を開始
  - ノードpで
    $succ(p+2^{i-1}) \leqq k < succ(p+2^i)$
    となるiをFinger tableで見つける
  - 実ノードsucc $(p+2^{i-1})$から探索を続ける

階層的アプローチ



- 全体のネットワークをドメインに分割，各ドメインは複数のサブドメインに分割，最下位のドメインはリーフドメインとなる
- 各ドメインDはドメイン内のエンティティを記録するディレクトリノードdir(D)を持つ
- リーフドメインR内のエンティティEのアドレスは，dir(R)の位置レコードに記録される
- リーフドメインRの親ドメインには，子ドメインへのポインタを位置レコードに記録する
- エンティティEの探索には，Eの位置レコードを記録しているドメインまで遡り，見つかれば，子ドメインへのポインタを辿ってリーフノードへ到達する

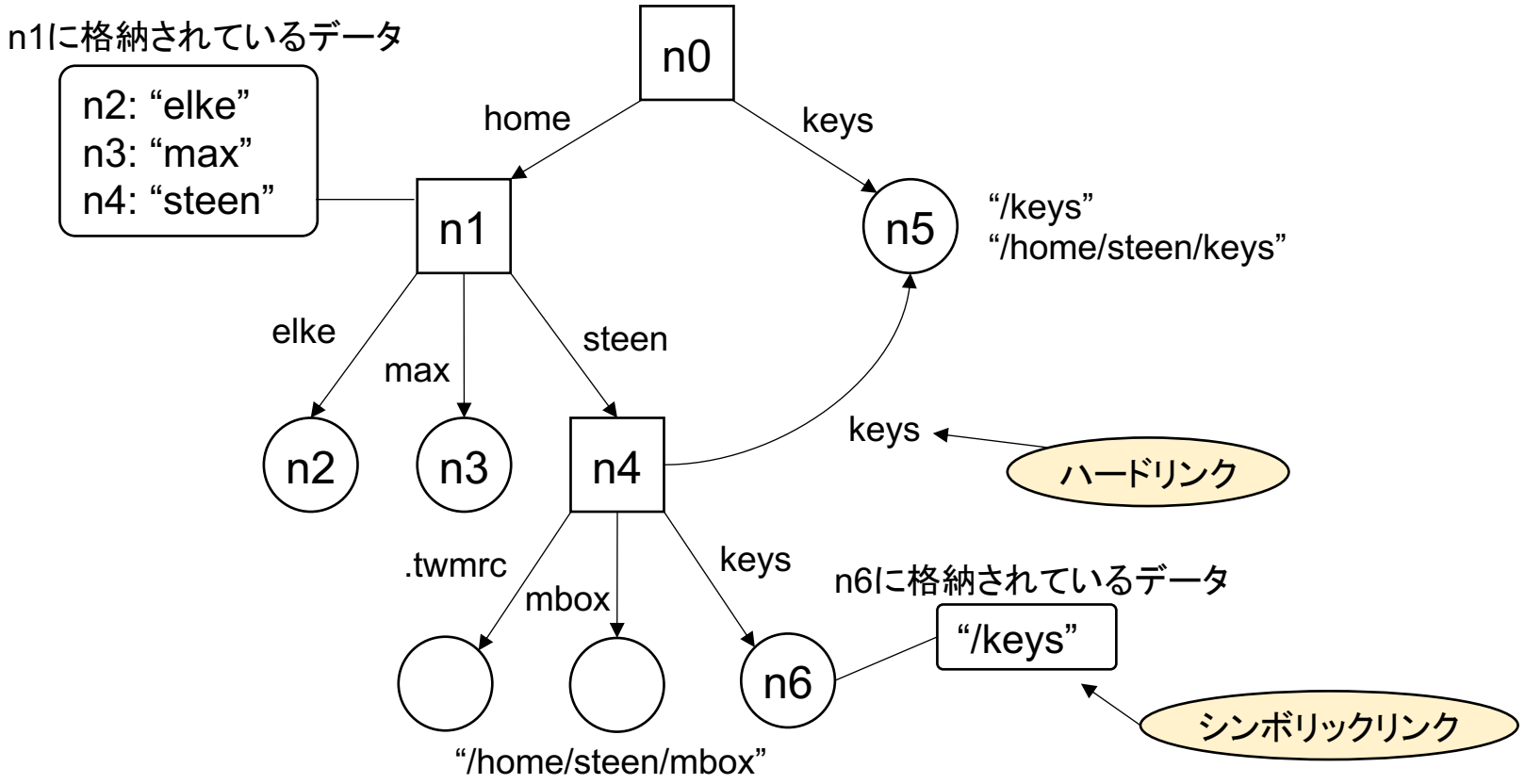
構造化された名前付け

- 名前は「名前空間」で管理される
- 名前空間は，以下の２種類のノードをもつラベル付き有向グラフ(「名前グラフ」)で表現される

  リーフノード(leaf node)： 名前付けされ，属性値を持つエンティティを表す

  ディレクトリノード(directory node)：１つ以上の出力枝を持ち，それぞれの枝は名前によりラベル付けされる
- 出力枝のみを持ち，入力枝を持たないノードはルートノード(root node)と呼ばれる

名前グラフの例



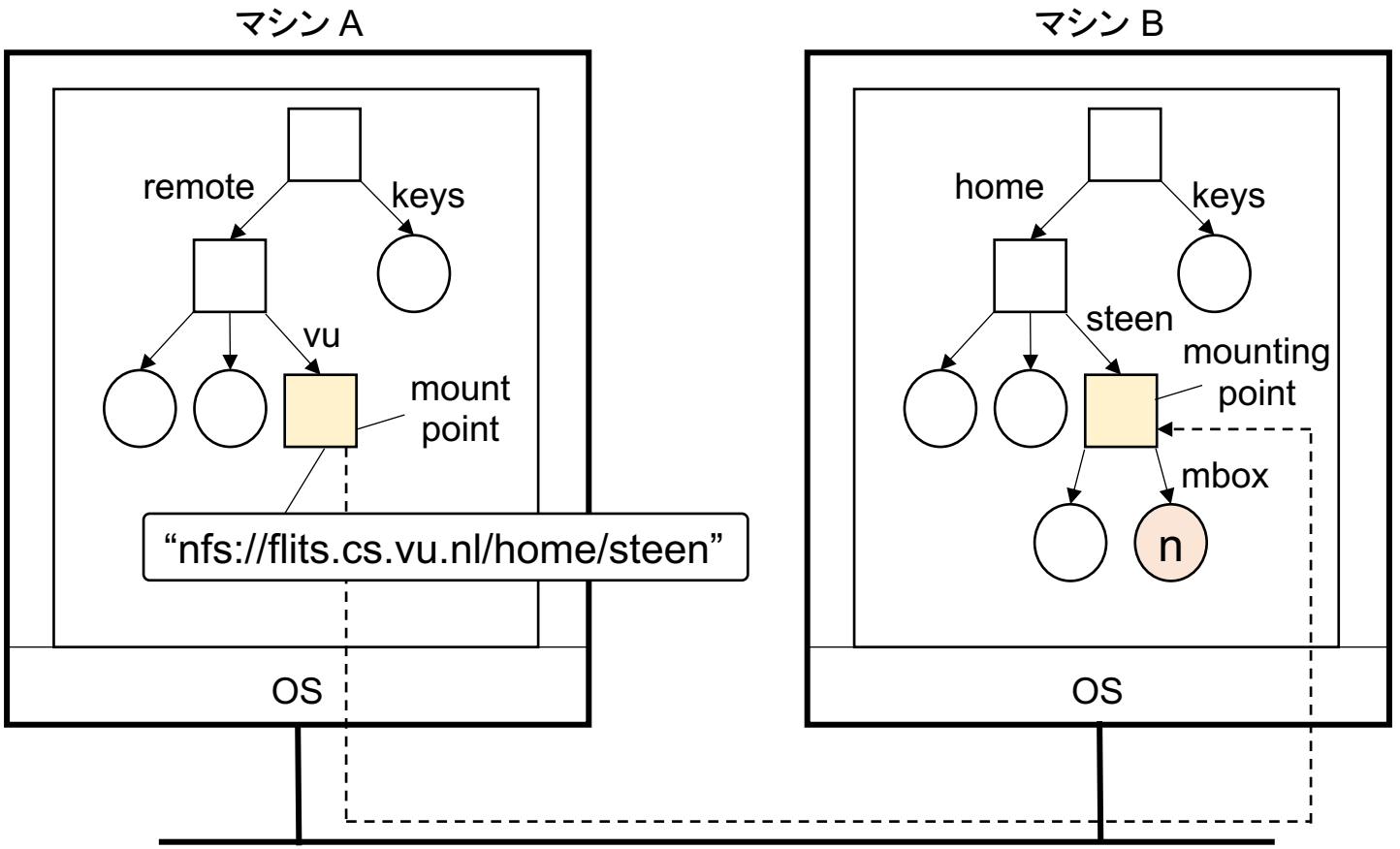
パス名は N:<ラベル-1, ラベル-2, …, ラベル-n>で表され，パス中の枝に対応するラベルの並びを参照する

名前解決は，パス中のラベル名を辿ることで実行される

リンクとマウント

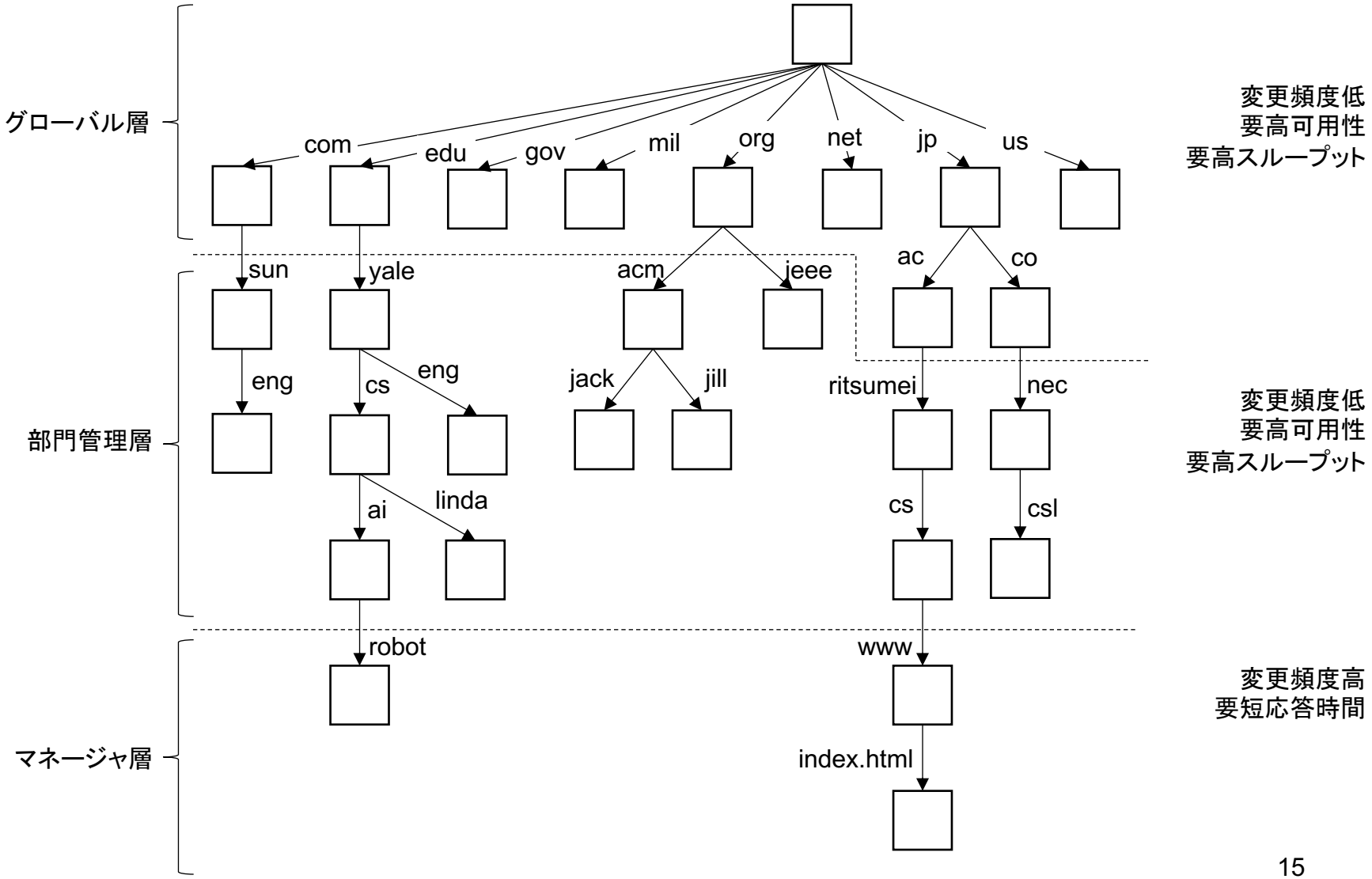
- エイリアス（alias）とは，同じエンティティの別名であり，以下の2つの方法で実現される
  - ハードリンク（hard link）：異なるパス名で同じノードを参照
  - シンボリックリンク（Symbolic link）：リーフノードに参照ノードの絶対パス名を格納
- ディレクトリノードに，異なる名前空間のディレクトリノードの識別子を格納することにより，異なる名前空間を併合する（マウント（mount））
  - マウントポイント：ノードの識別子を格納しているディレクトリノード
  - マウンティングポイント：異なる名前空間に存在しているディレクトリノード

Network File System (NFS)

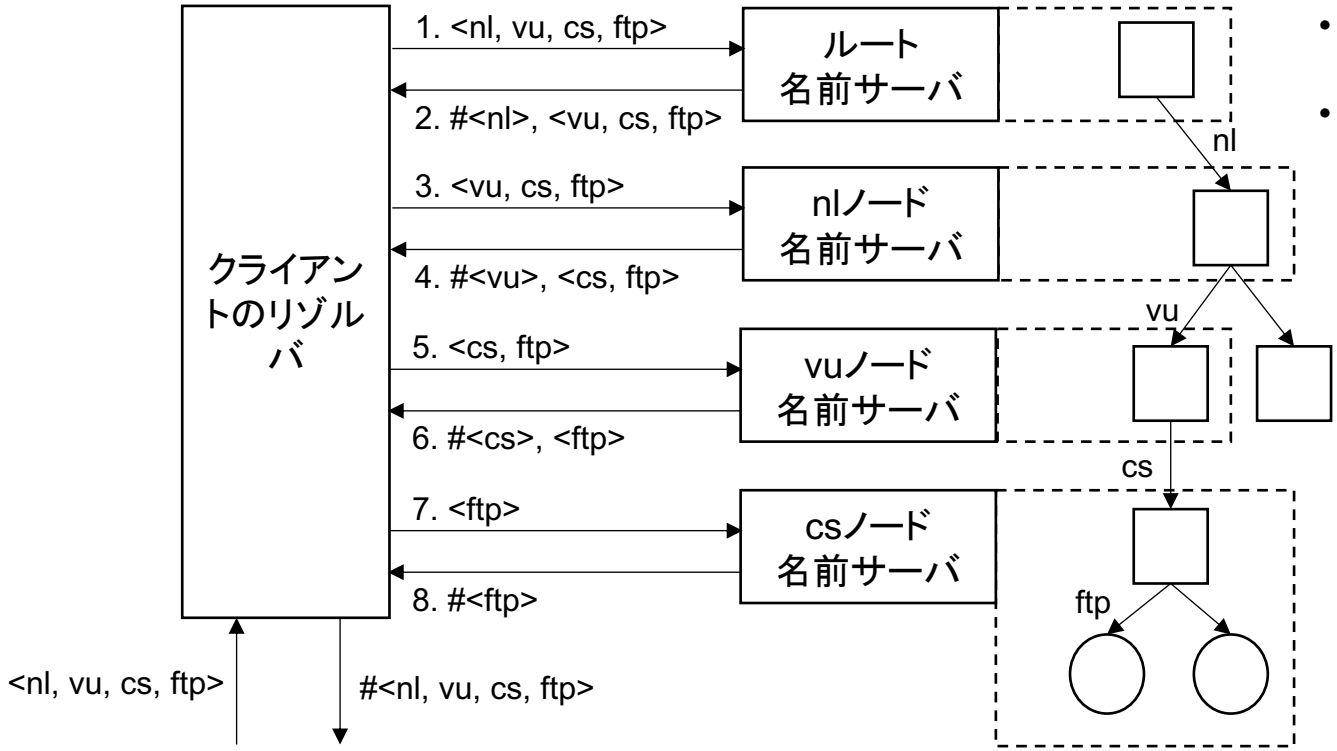


マシンBのノードnは，マシンAから /remote/vu/mbox という名前でアクセス可能
（ノードnがあたかもマシンAに存在しているようにユーザには見える）

名前空間の分散管理

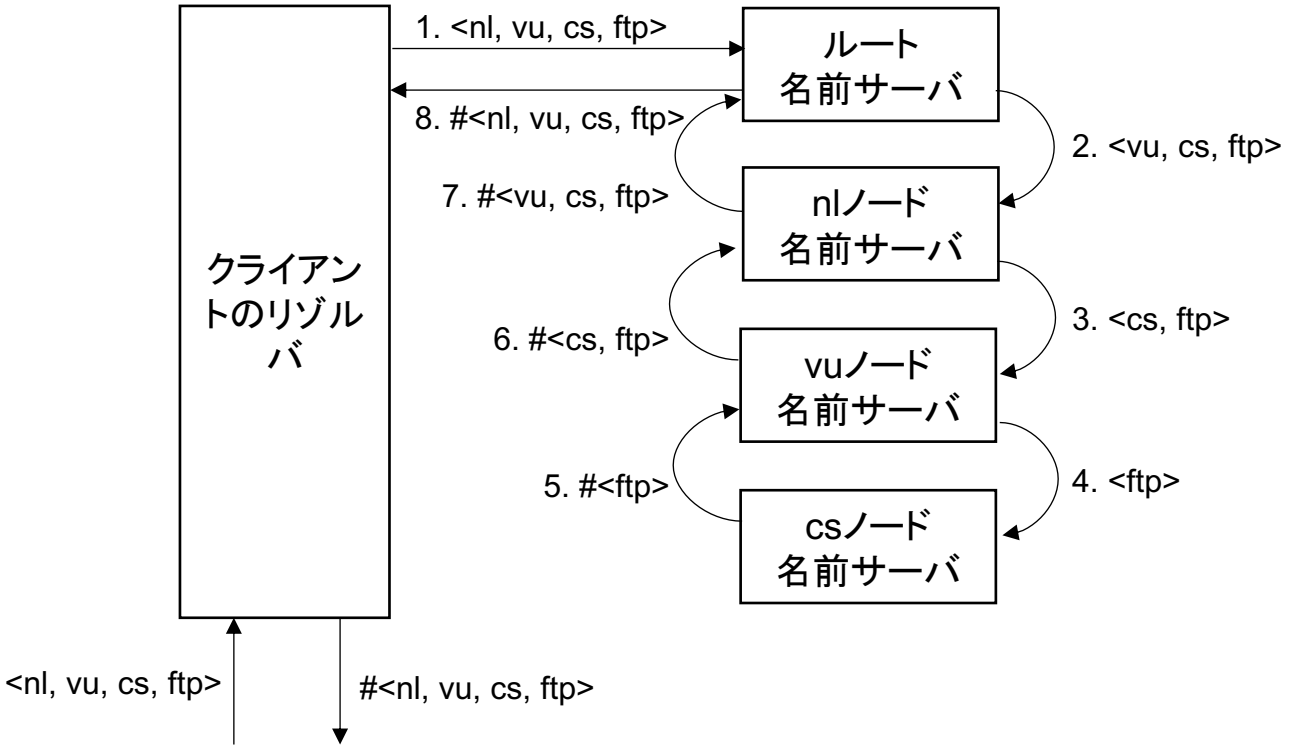

グローバル層
部門管理層
マネージャ層
com
edu
gov
mil
org
net
jp
us
sun
yale
acm
ieee
ac
co
eng
cs
eng
jack
jill
ritsumei
nec
ai
linda
cs
csl
robot
www
index.html
変更頻度低
要高可用性
要高スループット
変更頻度低
要高可用性
要高スループット
変更頻度高
要短応答時間

反復名前解決（Iterative name resolution）



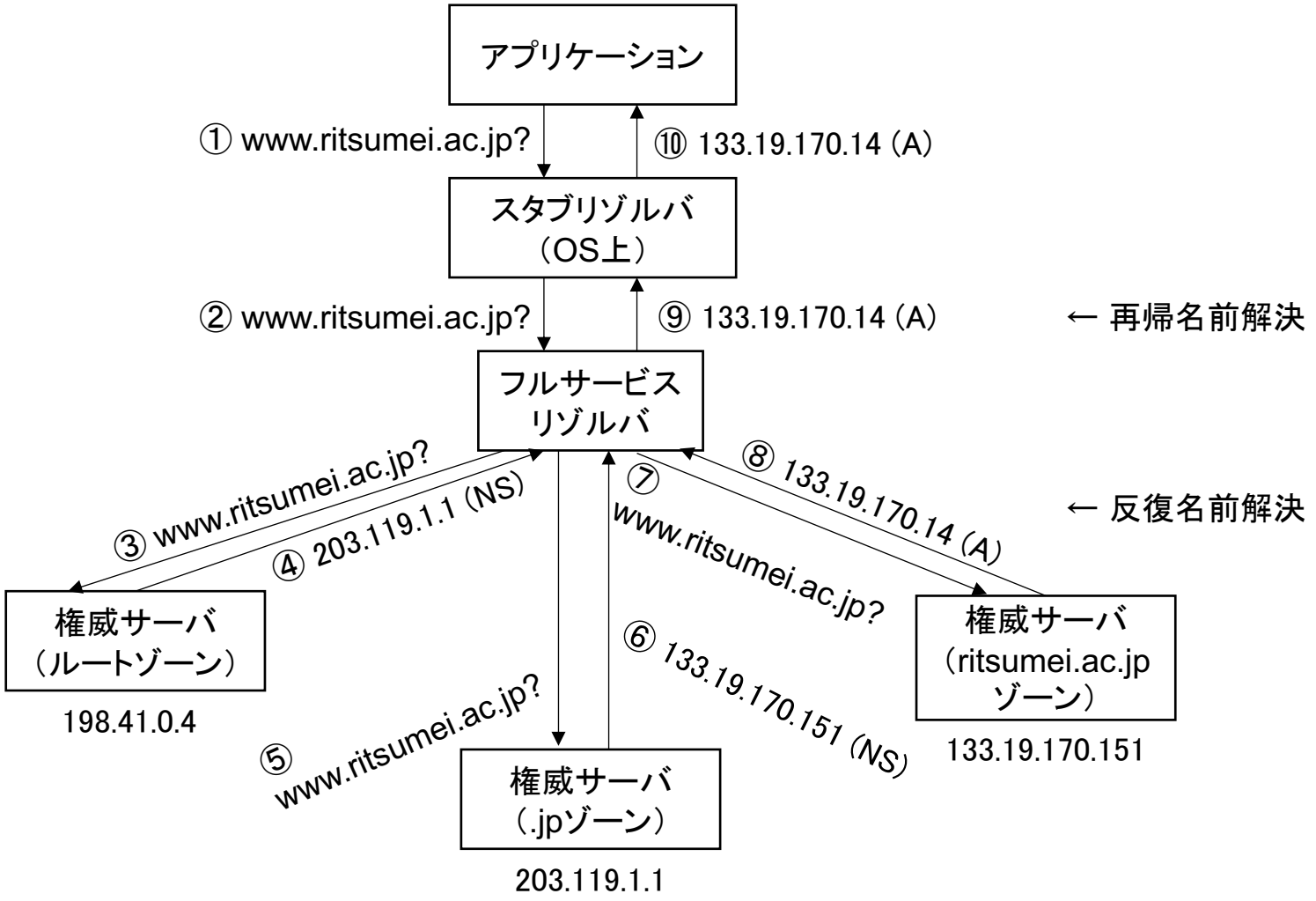
- クライアントのリゾルバがルート名前サーバに完全なパス名を渡す
- ルート名前サーバはできる限りパス名を解釈し，次に問い合わせるべき名前サーバのアドレスと，未解決のパス名をクライアントに返す
- クライアントは次の名前サーバに未解決のパス名を渡す。これをパス名が解決されるまで繰り返す

再帰名前解決（Recursive name resolution）



- 名前サーバ自身が次の名前サーバに結果を渡す
- 反復名前解決に対して以下の利点を持つ
  - 途中の名前サーバでのキャッシュが有効に働く
  - クライアントと名前サーバが離れている場合に，通信コストが低下する

The Domain Name System (DNS)



digコマンド

再帰名前解決

```
$ dig www.ritsumei.ac.jp

;; QUESTION SECTION:
;www.ritsumei.ac.jp. IN A

;; ANSWER SECTION:
www.ritsumei.ac.jp. 7896 IN A 133.19.170.14
```

反復名前解決

```
$ dig +norec @198.41.0.4 www.ritsumei.ac.jp
;; QUESTION SECTION:
;www.ritsumei.ac.jp.     IN          A
;; AUTHORITY SECTION:
jp.          172800      IN          NS          a.dns.jp.
;; ADDITIONAL SECTION:
a.dns.jp.    172800      IN          A           203.119.1.1

$ dig +norec @203.119.1.1 www.ritsumei.ac.jp
;; QUESTION SECTION:
;www.ritsumei.ac.jp.     IN          A
;; AUTHORITY SECTION:
ritsumei.ac.jp.    86400 IN          NS      rundz1151.ritsumei.ac.jp.
;; ADDITIONAL SECTION:
rundz1151.ritsumei.ac.jp. 86400      IN          A    133.19.170.151

$ dig +norec @133.19.170.151 www.ritsumei.ac.jp
;; QUESTION SECTION:
;www.ritsumei.ac.jp.                IN          A
;; ANSWER SECTION:
www.ritsumei.ac.jp.      10800      IN          A    133.19.170.14
```

属性ベース名前付け

- 一般にディレクトリサービスと呼ばれている
  - エンティティは（属性，値）の集合を持つ
  - 属性の値によりエンティティを検索できる
  - （例）電子メイルの場合，メイルは「送信者」「受信者」「件名」などの属性を持つ
- リソース記述フレームワーク（Resource Description Framework, RDF）
  - 各エンティティの属性値を統一的に管理ための仕組み
  - エンティティに対応する各リソースを（主語，述語，目的語）で記述
  - （例）（Person, name, Alice）は「名前（name）」が「Alice」であるような「人（Person）」であることを示す

LDAP (Lightweight Directory Access Protocol)

- 属性ベース名前付けと構造化した名前付けを組み合わせたディレクトリサービス
- （attribute, value）で各レコードを記述

Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Comp. Sc.
CommonName	CN	Main server
Mail_Servers	-	137.37.20.3, 130.37.24.6, 137.37.20.10
FTP_Server	-	130.37.20.20
WWW_Server	-	130.37.20.20

- 名前解決は大局的に一意な名前を用いて行われる

/C=NL/O=Vrije Universiteit/OU=Comp. Sc./CN=Main_Server