High Availability IRR

Yasuhiro Shirasaki

Tomoya Yoshida

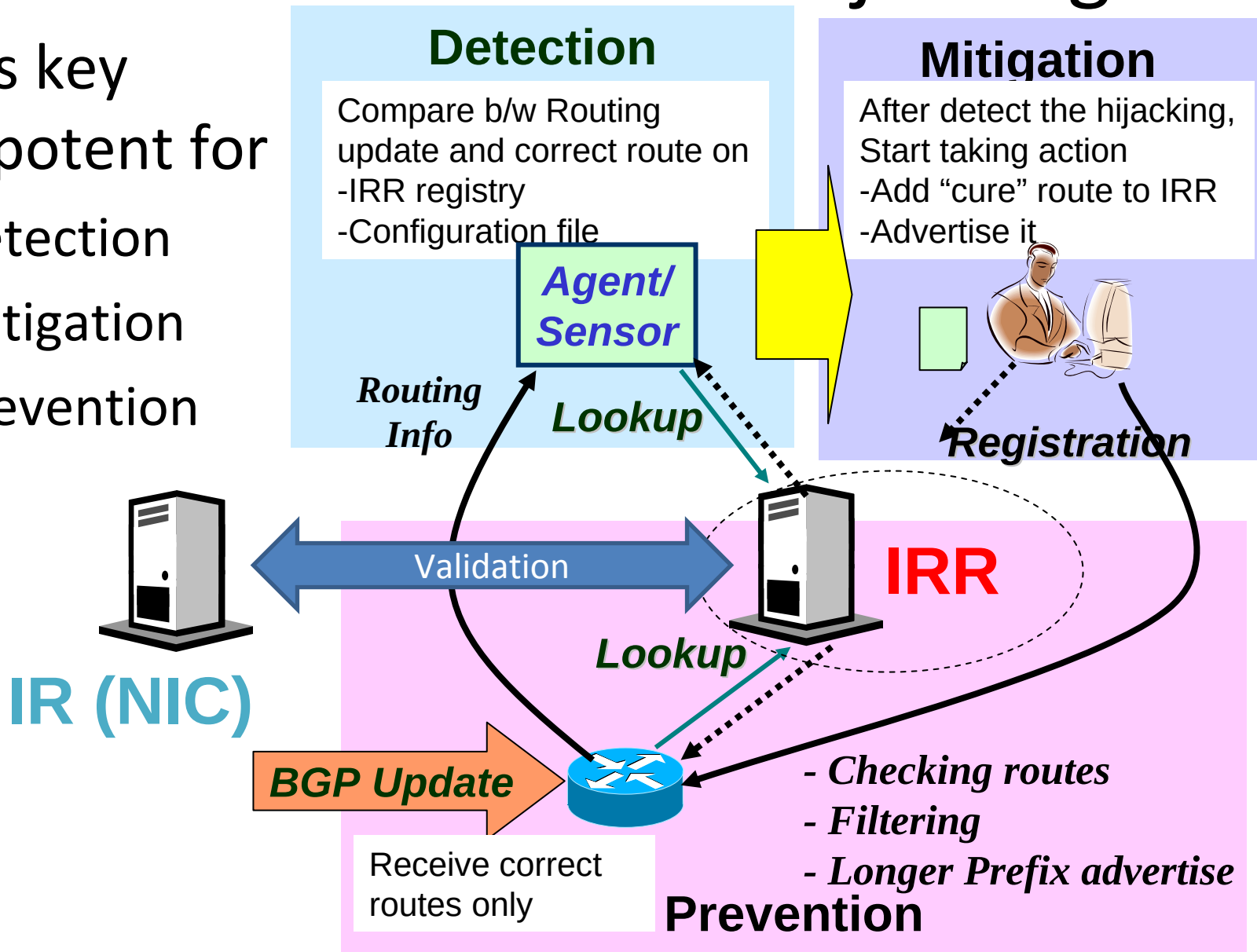
Anti Prefix Hijacking Project / NTT Communications

Why High Availability?

- Anti Prefix hijacking system in JP
  - Real time validation with JPIRR
  - Registration check with IR (JPNIC)
- To deploy “IRR based Anti prefix hijacking”
  - Redundancy (Availability)
  - Valid IRR entry (Integrity)
  - Performance
- Research Project on Anti Prefix Hijacking
  - Japanese Government (Ministry of Internal Affairs and Communications) research project
  - to develop detection/mitigation/prevention mechanism

IRR based Anti Prefix Hijacking

- IRR is key competent for
  - Detection
  - Mitigation
  - Prevention



Current IRR systems

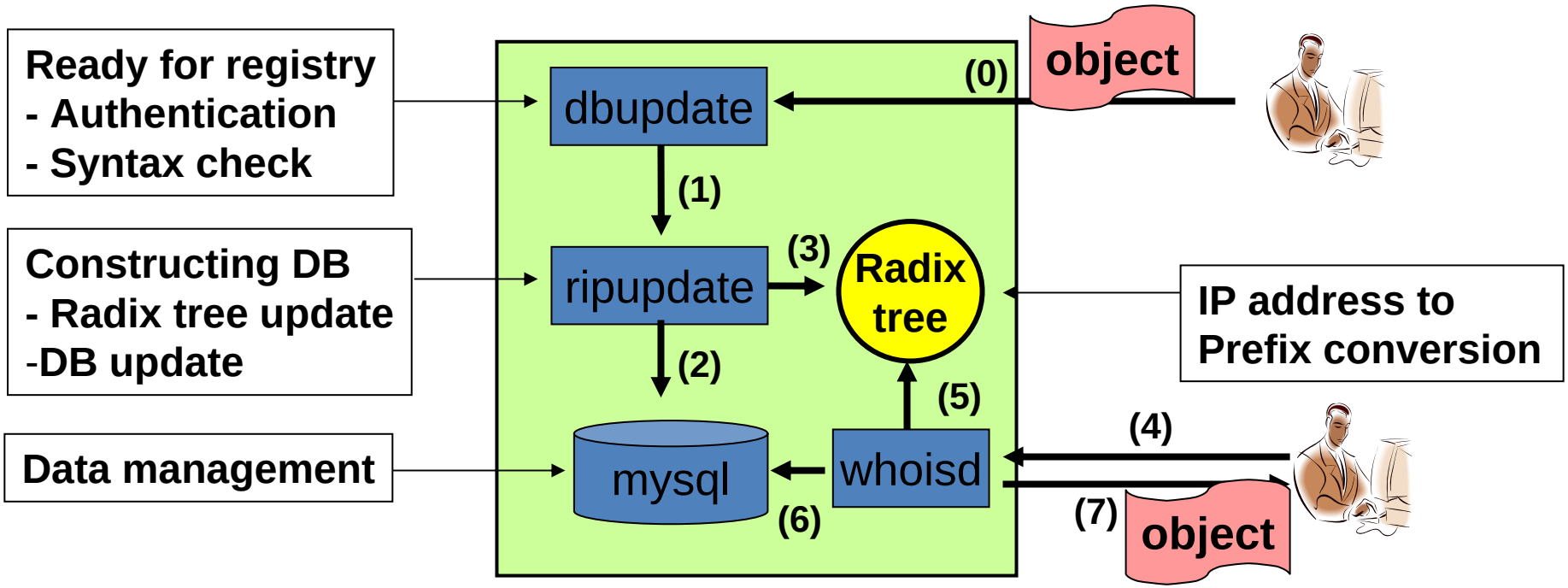
- RIPE whoisd — RIPE, APNIC, AfriNIC
- Merit IRRd — RADB, JPIRR, VRR, other ISP's IRR

	RIPE Whoisd	*Merit IRRd*
Structure	**Modularized**	**Monolithic**
Data management	**RDBMS (MySQL)**	**Text file**
Object registry	**Mail, Web (other tool)**	**Mail, Web (other tool)**
Mirroring protocol	**NRTM**	**NRTM**
RPSL correspondense	**RPSLng (RFC4012)**	**RPSLng (RFC4012)**
Error check	**Strict (Sequence check)**	**Loose**
System Scalability	**Yes (RDBMS)**	**No**
Latest version	**Active Dev on CVS**	**Irrd-2.3.4 2007/6/11**

HA IRR Software Development

- We have picked up RIPE whois-server as the base software
- Now we're working on
  - Redundancy
  - Down-time minimization
  - High performance
    - Especially response time and scalability
  - Ability to handle "signed" data objects

RIPE whoisd

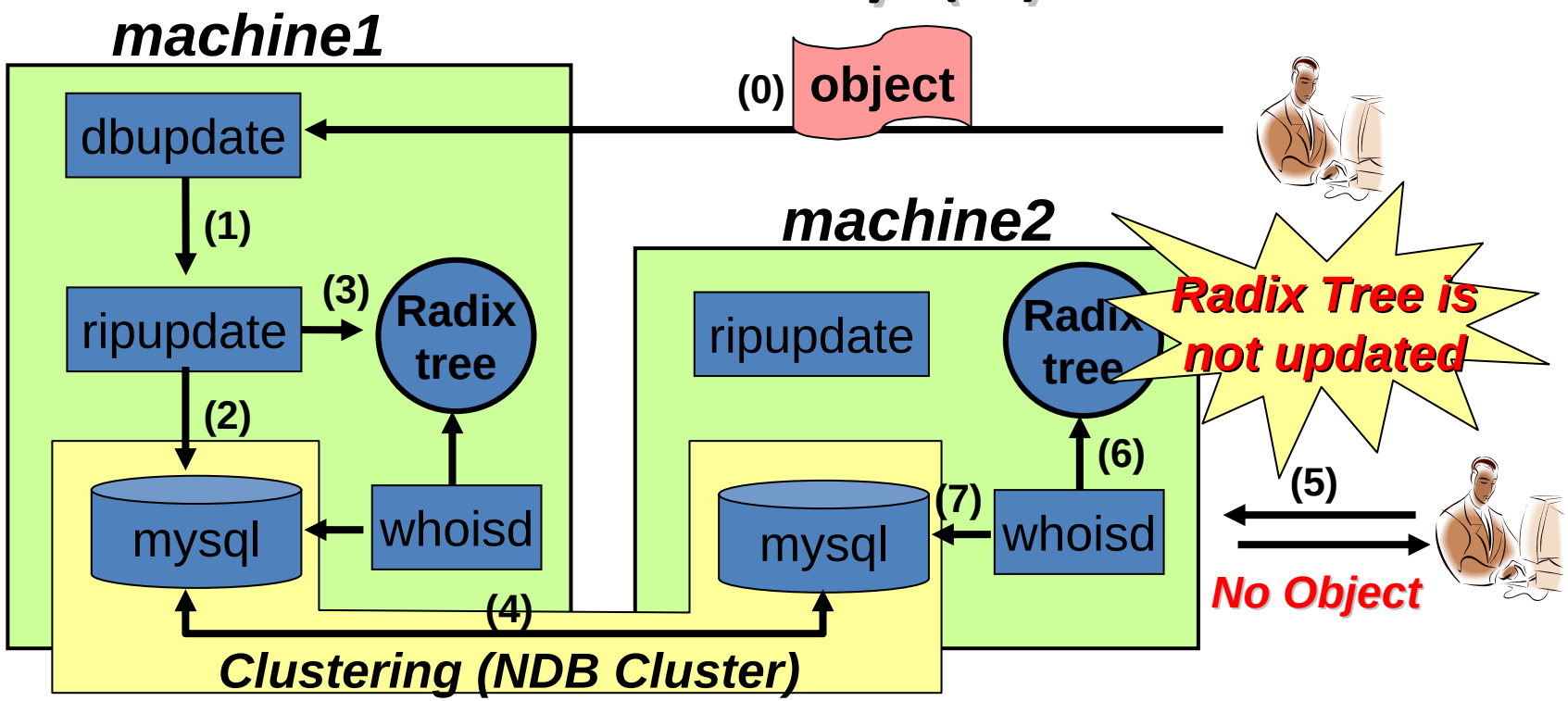


(0) Object registration by Operator
(1) Authentication and
(2) Update “MySQL Database”
(3) Constructing “Radix Tree”
(4) Whois query
(5) Check address (IP Prefix check, cache)
(6) Database search
(7) Whois reply

Registry Process

Lookup Process

Redundancy (1)



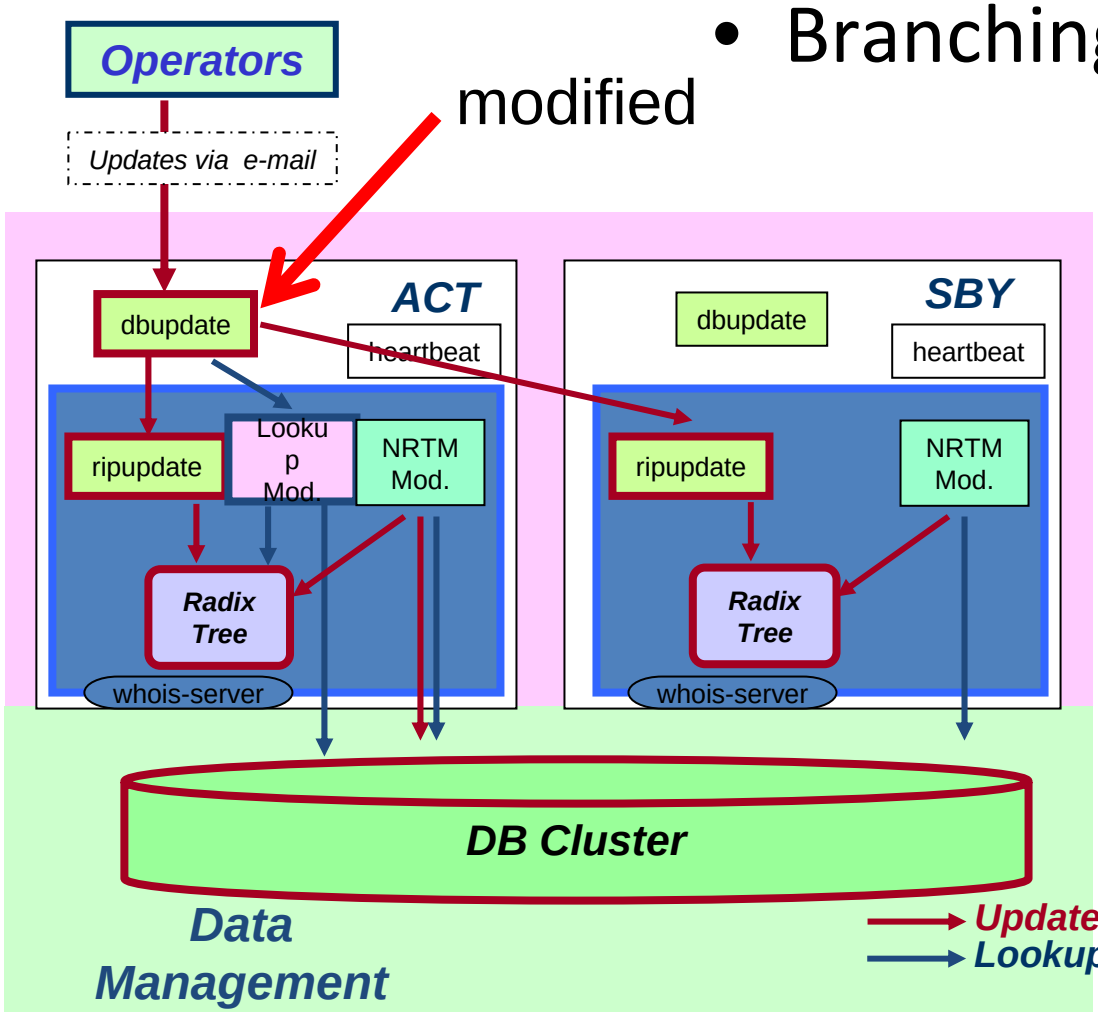
(0) Object registry by Operator
(1) Send Object to ripupdate for Database constructing
(2) Update “MySQL Database”
(3) Constructing “Radix Tree”
(4) Database sync
(5) Whois query
(6) Check address (IP Prefix check)
(7) Database search

Registry Process

Lookup Process

Redundancy (2) Type-A

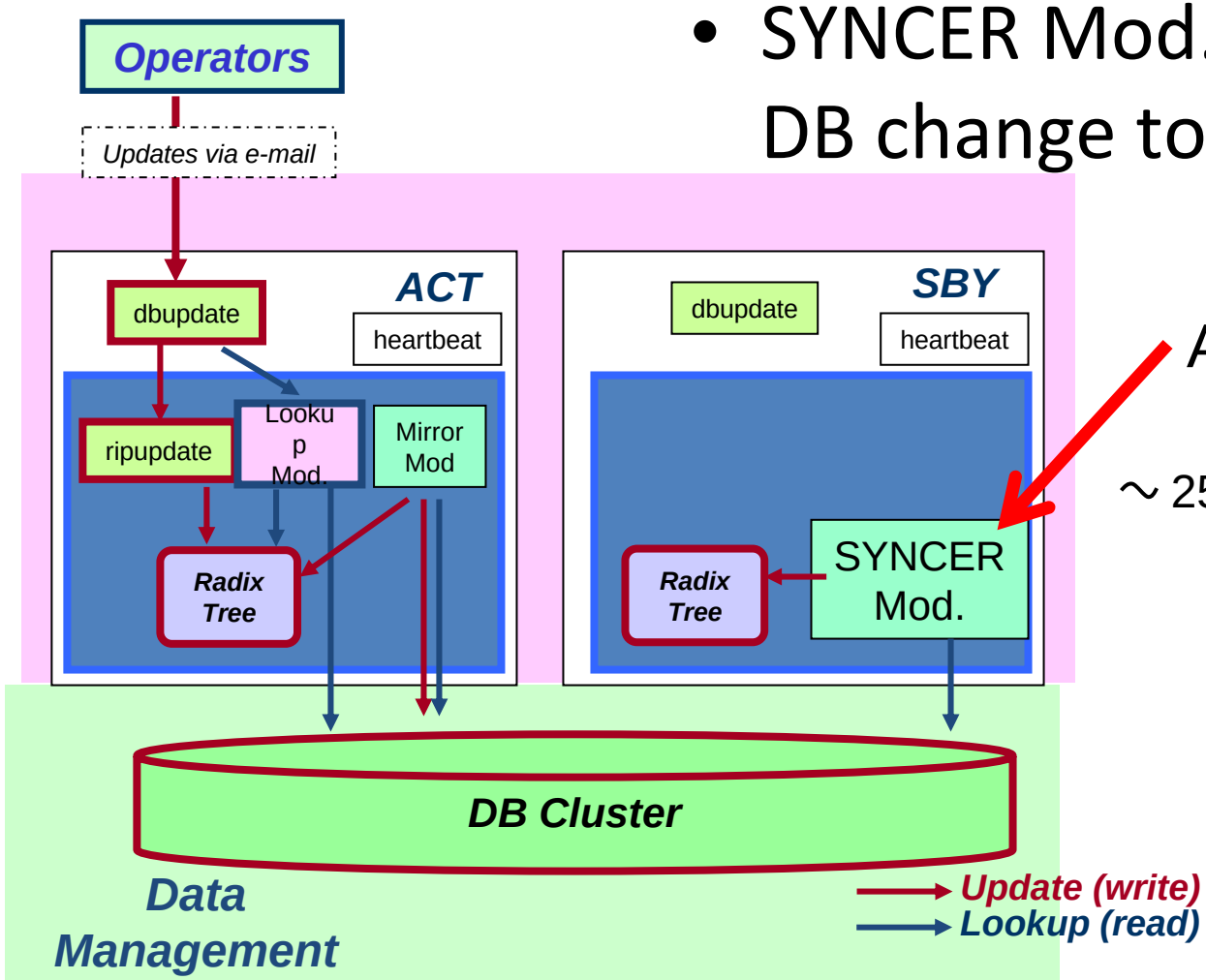
- Branching at dbupdate



∼500 lines modification

Redundancy (3) Type-B

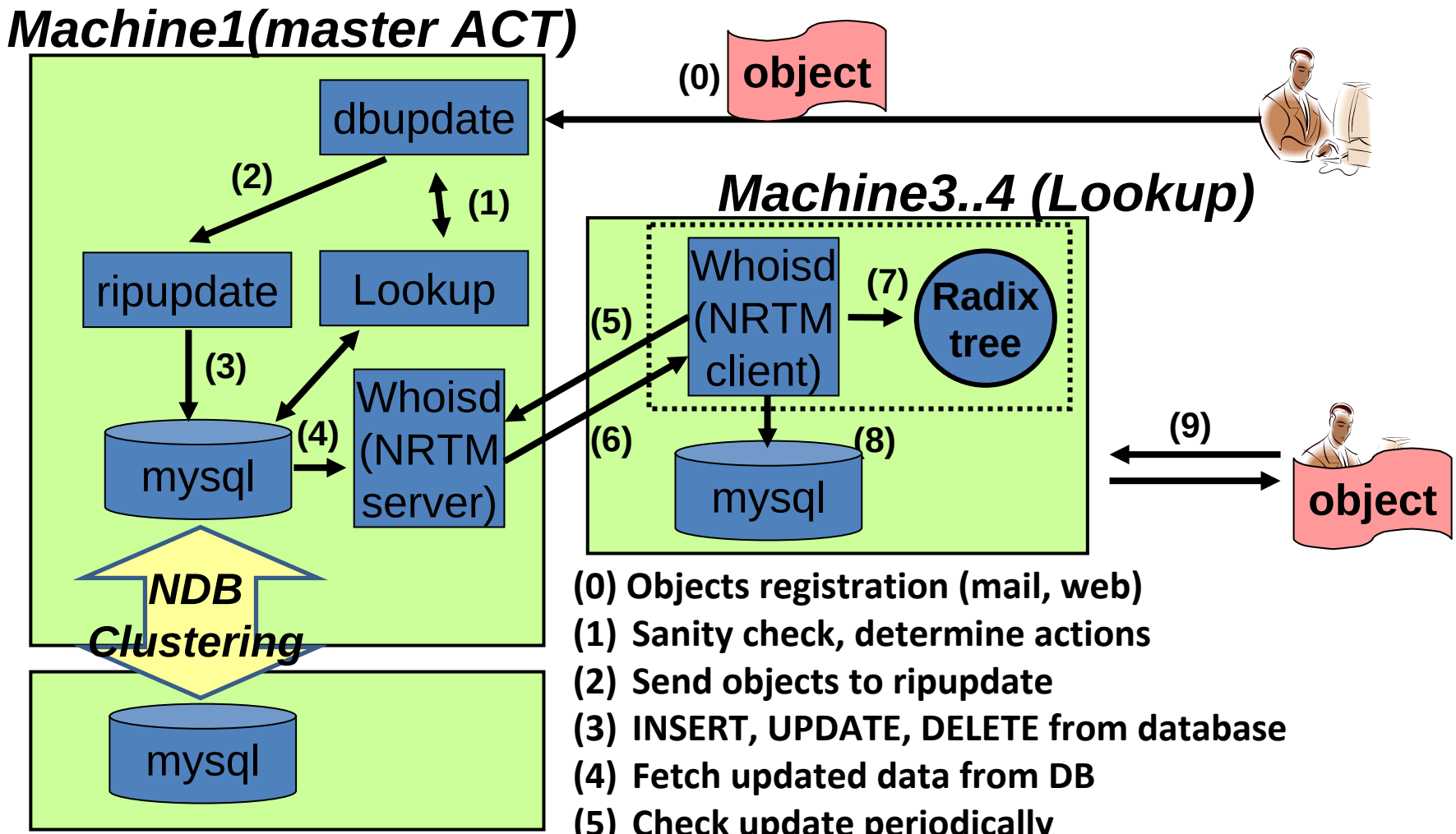
- SYNCER Mod. reflects DB change to Radix Tree



Added

～2500 lines modification

Redundancy (4)

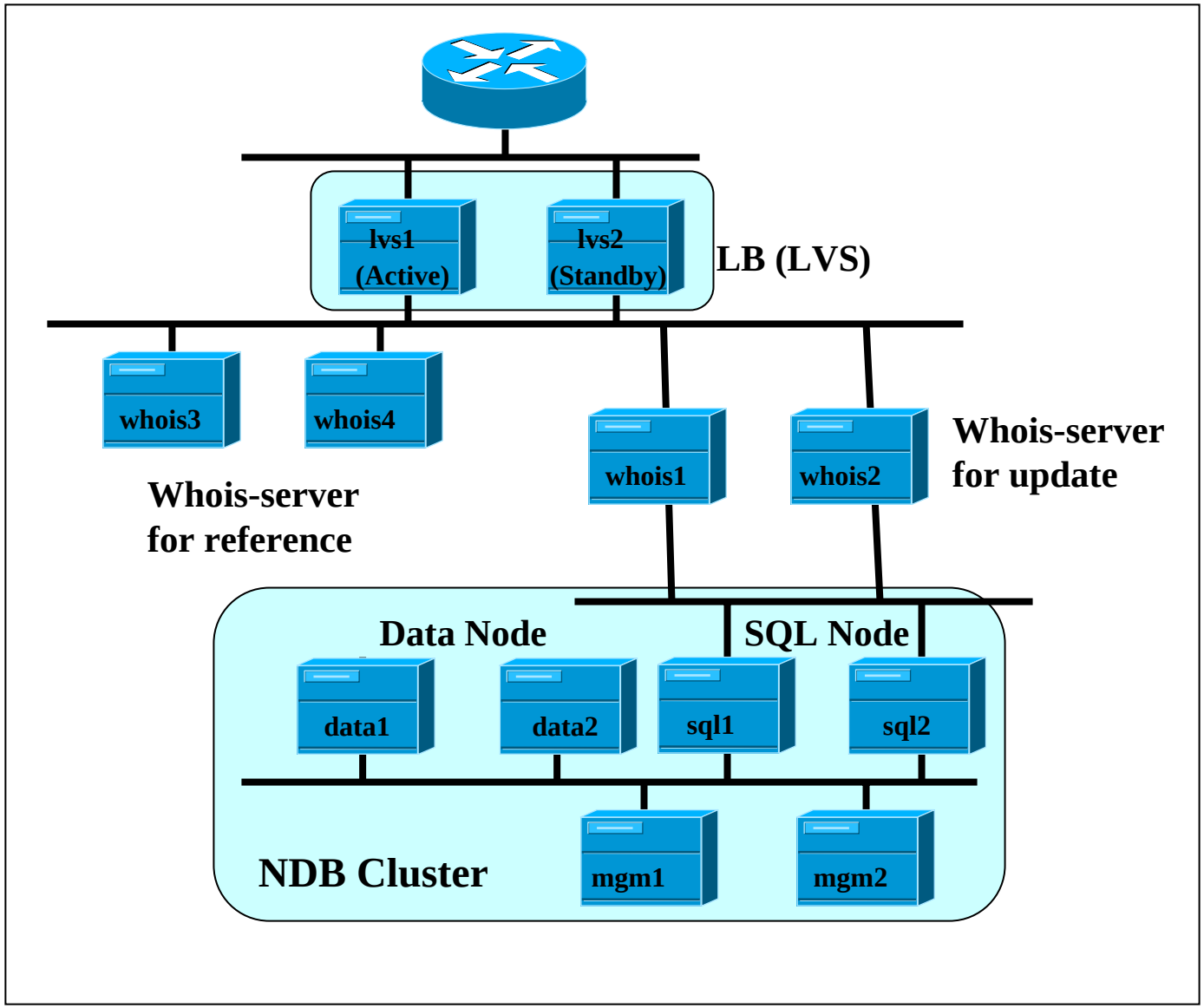


(0) Objects registration (mail, web)
(1) Sanity check, determine actions
(2) Send objects to ripupdate
(3) INSERT, UPDATE, DELETE from database
(4) Fetch updated data from DB
(5) Check update periodically
(6) Send updates to NRTM client
(7) Update Radix Tree
(8) Updates Radix Tree and store data into local DB
(9) Reply for users' queries

Down time minimization

- Radix Tree (RX) rebuilding issue
  - Rebuilding RX in a boot sequence takes more than 5 minutes for RIPE-DB
  - Service interruption by blocking DB updates while rebuilding RX
  - NDBCLUSTER storage engine doesn’t support Multi Version Concurrency Control (MVCC) mechanism
    - No “snapshot”
    - Future planned
- Current Workaround
  - Make a copy of table in MEMORY storage from slow NDBCLUSTER storage to minimize service interruption

Test system configuration



Performance target

- Registry
  - RADB
    - 188717 objects
    - ~240 actions/day (0.13% of total object number)
      - 117 days (From 1st Mar to 25th Jun)
      - 27805 sequences change (455871 ---> 483676)
  - RIPE
    - 2461141 objects (including IR objects)
    - ~7000 actions/day (0.28% of total object number)
      - 133 days (From 6th Feb to 19th Jun)
      - 925890 sequences change (8701661 ---> 9627551)
    - 240 updates/minutes at peak
- Lookup
  - RADB queries
    - About less than 4M queries/day

From http://www.radb.net/stats-history.html

Remaining Items

- Performance
  - Memory issue

    4Gbytes memory is not enough for NDB cluster

    ---> Upgrade configuration (amd64/8GB～)
  - Scalability
    - Clustering with many servers
- Redundancy
  - Off-site redundancy
- Field trial
  - Redundancy test b/w Tokyo and Osaka

Thank you

Yasuhiro Shirasaki
yasuhiro@nttv6.jp

Tomoya Yoshida
yoshida@ocn.ad.jp