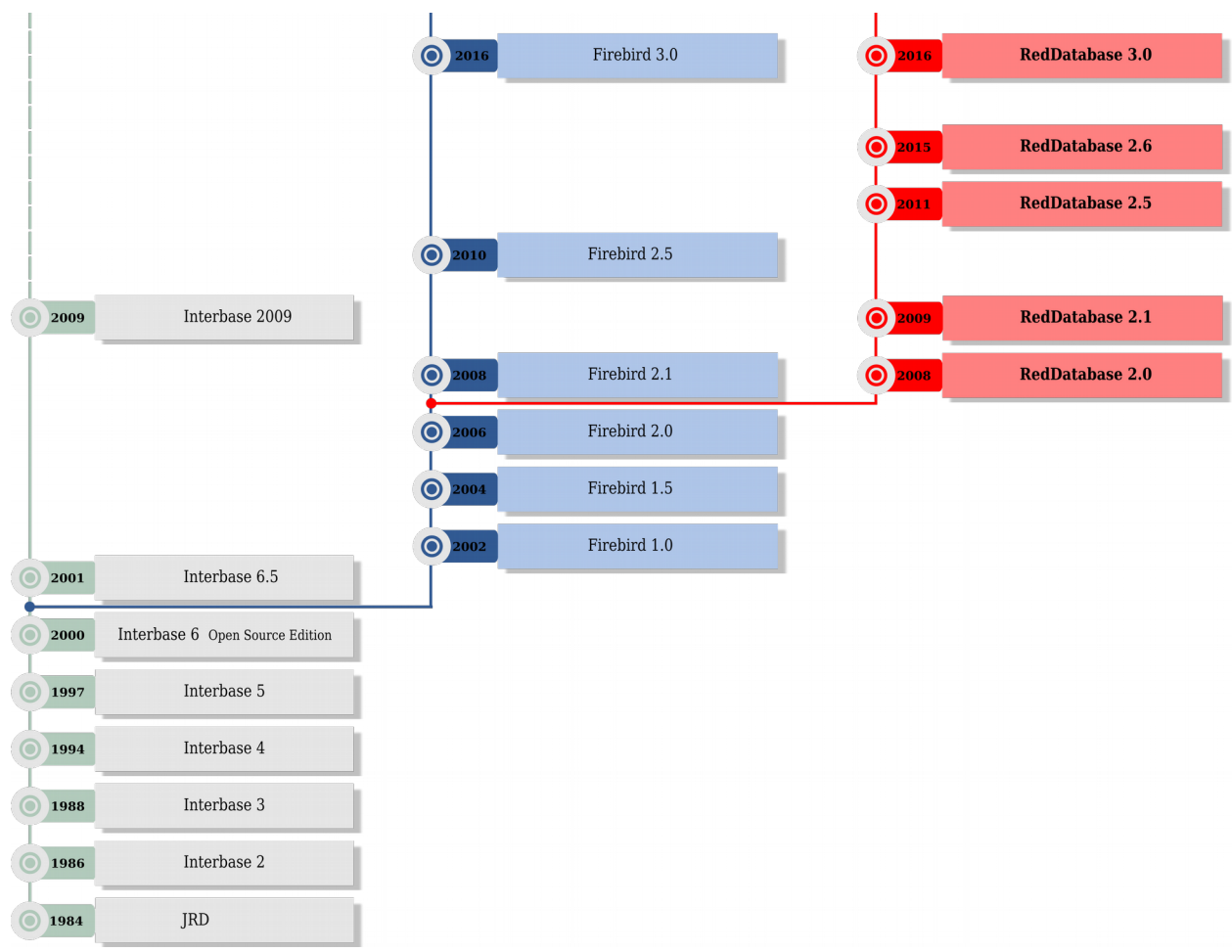




New features of Firebird ported from RedDatabase

Roman Simakov, director of system development department

History of Red Database



Security features

- Cryptographic plugin
- Multi-factor authentication
- Cumulative roles
- DML access control
- DDL access control
- Service access control
- Record filtering

Functional features

- Java Stored Procedures
- Full Text Search
- LDAP/AD integration
- StandBy cluster (engine-level replication)

Automated Information System of Federal Service for Officers of Justice of Russia

- AIS is installed and work in 85 regional departments and in the main office of FSOJ of Russia
- Total amount of Red Database installations are about 2720, i.e. every city of Russia has one or several Red Database servers
- AIS handles more than 10^9 documents per year
- AIS works in 24/7 mode
- Some databases more than 1TB and a lot of data goes to archived set of database files
- 100x of concurrent connections
- 100 000x transactions per hour



Example of load

```

roman.simakov@mvv-reestr:/opt/RedDatabase/bin
roman@roman-ubuntu: ~/prj/fb/firebird/gen/De... x
roman@roman-ubuntu: ~/prj/testgen x
roman.simakov@mvv-reestr:/opt/RedDatabase/... x

Доброе утро!
Сегодня в полдень вылетает поезд № 21 [[||||| 86.9%]]
Сегодня в полдень вылетает поезд № 22 [[||||| 85.4%]]
Сегодня в полдень вылетает поезд № 23 [[||||| 85.7%]]
Сегодня в полдень вылетает поезд № 24 [[||||| 82.7%]]
Сегодня в полдень вылетает поезд № 25 [[||||| 81.8%]]
Сегодня в полдень вылетает поезд № 26 [[||||| 78.5%]]
Сегодня в полдень вылетает поезд № 27 [[||||| 77.4%]]
Сегодня в полдень вылетает поезд № 28 [[||||| 76.1%]]
Сегодня в полдень вылетает поезд № 29 [[||||| 74.7%]]
Сегодня в полдень вылетает поезд № 30 [[||||| 72.6%]]
Сегодня в полдень вылетает поезд № 31 [[||||| 72.3%]]
Сегодня в полдень вылетает поезд № 32 [[||||| 70.7%]]
Сегодня в полдень вылетает поезд № 33 [[||||| 66.5%]]
Сегодня в полдень вылетает поезд № 34 [[||||| 63.9%]]
Сегодня в полдень вылетает поезд № 35 [[||||| 63.9%]]
Сегодня в полдень вылетает поезд № 36 [[||||| 54.8%]]
Сегодня в полдень вылетает поезд № 37 [[||||| 75.9%]]
Сегодня в полдень вылетает поезд № 38 [[||||| 100.0%]]
Сегодня в полдень вылетает поезд № 39 [[||||| 100.0%]]
Сегодня в полдень вылетает поезд № 40 [[||||| 69.9%]]

Mem [[||||| 105974/516241MB]]
Swp [[||||| 263/524287MB]]

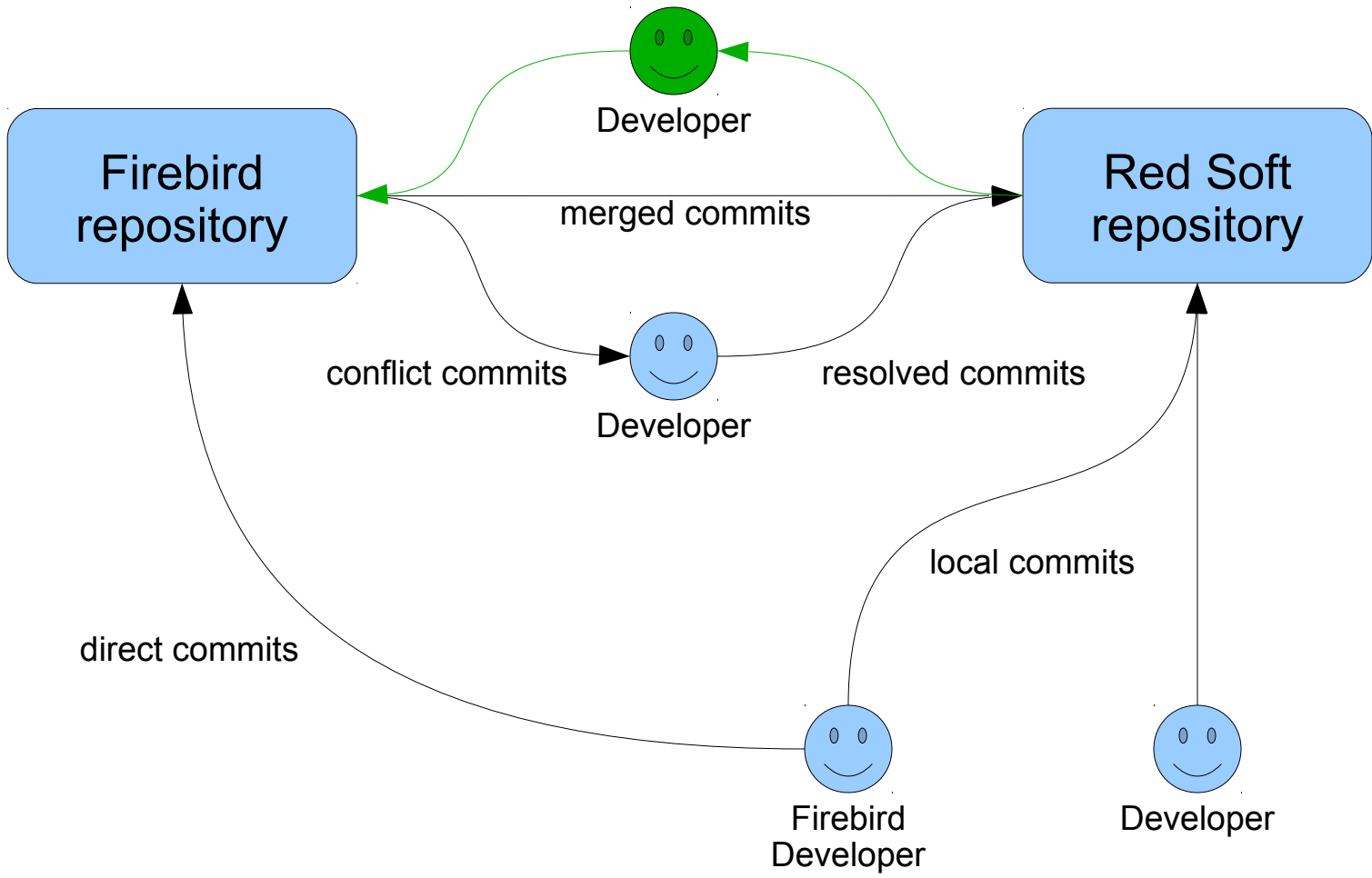
Tasks: 143, 958 thr; 6 running
Load average: 57.24, 36.27, 32.15
Uptime: 179 days(!), 21:40:27

Учетная запись для сайта и мобильного приложения Аэроэкспресс/Account for the Aeroexpress... - уважаемый пассажир! Для Вашего удобства автоматически создана учет

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
54964 root 20 0 593M 292M 73152 R 101.0 0.1 1h02:27 /opt/RedDatabase/bin/rdb_inet_server
70106 root 20 0 468M 167M 73040 S 101.0 0.0 14h13:52 /opt/RedDatabase/bin/rdb_inet_server
60060 root 20 0 468M 167M 73040 R 101.0 0.0 8:46:51 /opt/RedDatabase/bin/rdb_inet_server
70455 root 20 0 593M 292M 73152 S 100.0 0.1 16h35:09 /opt/RedDatabase/bin/rdb_inet_server
60918 root 20 0 455M 154M 73164 R 100.0 0.0 0:32:89 /opt/RedDatabase/bin/rdb_inet_server
70475 root 20 0 455M 154M 73164 S 100.0 0.0 13h14:24 /opt/RedDatabase/bin/rdb_inet_server
70294 root 20 0 505M 172M 73096 S 82.0 0.0 14h34:29 /opt/RedDatabase/bin/rdb_inet_server
60600 root 20 0 505M 172M 73096 R 82.0 0.0 2:50:87 /opt/RedDatabase/bin/rdb_inet_server
37613 root 20 0 427M 104M 48744 S 25.0 0.0 5:59:43 /opt/RedDatabase/bin/rdb_inet_server
60167 root 20 0 427M 104M 48744 S 24.0 0.0 0:40:32 /opt/RedDatabase/bin/rdb_inet_server
69466 t-mvv 20 0 61.8G 11.0G 20436 S 13.0 2.2 43h07:06 /usr/java/default/jre/bin/java -Djava.util.logging.config.file=/mvv/tom
53646 root 20 0 406M 85496 49316 S 8.0 0.0 1:42:51 /opt/RedDatabase/bin/rdb_inet_server
41303 root 20 0 324M 67232 48728 S 8.0 0.0 4:01:25 /opt/RedDatabase/bin/rdb_inet_server
60780 root 20 0 324M 67232 48728 S 8.0 0.0 0:06:46 /opt/RedDatabase/bin/rdb_inet_server
60694 root 20 0 406M 85496 49316 R 6.0 0.0 0:04:28 /opt/RedDatabase/bin/rdb_inet_server
60982 roman.sim 20 6 111M 3148 1272 R 5.0 0.0 0:00:82 htop

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
  
```

Development process (schema)



DDL access control (CORE-735 since 2003)

Problem: Previously **anyone** who connected to database could create a table

Syntax

```
GRANT CREATE <OBJECT> TO [USER | ROLE] <user/role name> [WITH GRANT OPTION]
GRANT ALTER ANY <OBJECT> TO [USER | ROLE] <user/role name> [WITH GRANT OPTION]
GRANT DROP ANY <OBJECT> TO [USER | ROLE] <user/role name> [WITH GRANT OPTION]
REVOKE [GRANT OPTION FOR] CREATE <OBJECT> FROM [USER | ROLE] <user/role name>
REVOKE [GRANT OPTION FOR] ALTER ANY <OBJECT> FROM [USER | ROLE] <user/role name>
REVOKE [GRANT OPTION FOR] DROP ANY <OBJECT> FROM [USER | ROLE] <user/role name>
```

Where **OBJECT** can be:

TABLE, VIEW, PROCEDURE, FUNCTION, PACKAGE, GENERATOR, SEQUENCE, DOMAIN,
EXCEPTION, ROLE, CHARACTER SET, COLLATION, FILTER

DDL access control: Simple example

```
GRANT CREATE TABLE TO Joe
```

*ALTER possible because of Joe is owner of JoeT

```
GRANT ALTER ANY TABLE TO Joe
```

*Triggers and indices re-use table privileges

```
REVOKE CREATE TABLE FROM Joe
```

```
CREATE TABLE JoeT (I INTEGER)
```

```
ALTER TABLE JoeT ...
```

```
ALTER TABLE BobT ...
```

```
CREATE INDEX ON JoeT ...
```

```
CREATE INDEX ON BobT ...
```

```
CREATE TRIGGER ON T ...
```

```
ALTER TRIGGER ON T ...
```


DDL access control: special form for managing database

```
GRANT CREATE DATABASE TO [USER | ROLE] <user/role name>
GRANT ALTER DATABASE TO [USER | ROLE] <user/role name> [WITH GRANT OPTION]
GRANT DROP DATABASE TO [USER | ROLE] <user/role name> [WITH GRANT OPTION]
REVOKE CREATE DATABASE FROM [USER | ROLE] <user/role name>
REVOKE [GRANT OPTION FOR] ALTER DATABASE FROM [USER | ROLE] <user/role name>
REVOKE [GRANT OPTION FOR] DROP DATABASE FROM [USER | ROLE] <user/role name>
```

ALTER DATABASE permissions is used to check following actions:

- 1) Altering database itself
- 2) Commenting on database or db level triggers
- 3) Managing db level triggers
- 4) Managing shadows
- 5) Direct editing RDB\$FILES system table

Cumulative roles (CORE-1815 since 2008)

Now you can grant role to another role except circle references

Syntax:

```
GRANT [DEFAULT] <role name> TO [USER | ROLE] <user/role name> [WITH ADMIN OPTION]
REVOKE [DEFAULT] <role name> FROM [USER | ROLE] <user/role name> [WITH ADMIN OPTION]
```

ADMIN OPTION allows grantee to grant the role to another user or role

- 1) WORKER->MANAGER->Joe
- 2) WORKER->MANAGER=>Joe
- 3) WORKER=>MANAGER->Joe
- 4) WORKER=>MANAGER=>Joe

Syntax:

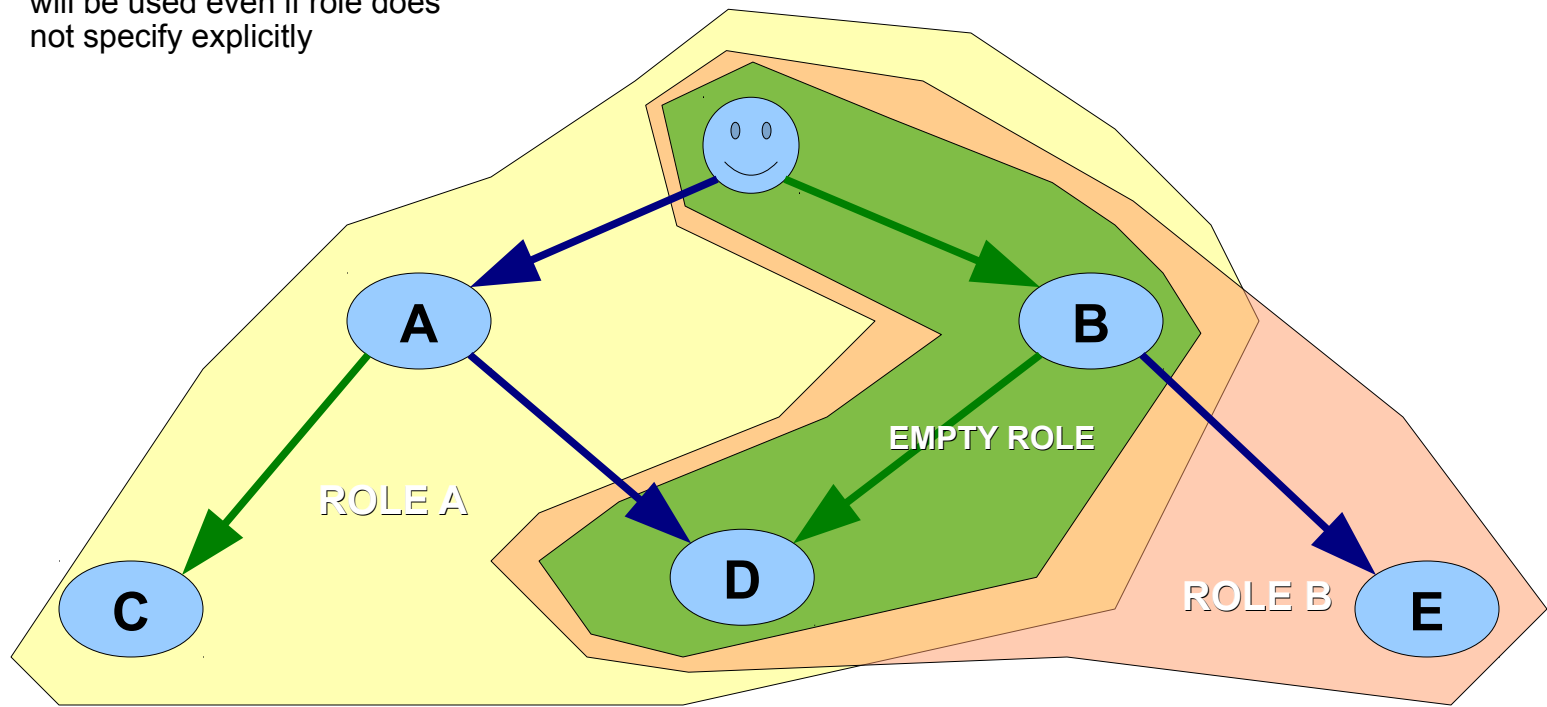
RDB\$ROLE_IN_USE(role_name varchar(32)) RETURNS BOOLEAN

To get a list of currently active roles you can run:

```
SELECT * FROM RDB$ROLES WHERE RDB$ROLE_IN_USE(RDB$ROLE_NAME)
```

Cumulative roles: DEFAULT explanation

DEFAULT means that this role will be used even if role does not specify explicitly



➔ DEFAULT

➔ WITHOUT DEFAULT

SQL SECURITY (SQL STANDARD 2003, 2011)

Syntax

```
CREATE TABLE <TABLENAME> (...) [SQL SECURITY {DEFINER | INVOKER}]

ALTER TABLE <TABLENAME> ... [{ALTER SQL SECURITY {DEFINER | INVOKER} | DROP SQL SECURITY}]

CREATE [OR ALTER] TRIGGER <TRIGGERNAME> ... [SQL SECURITY {DEFINER | INVOKER} | DROP SQL SECURITY]
[AS ...]

CREATE [OR ALTER] FUNCTION <FUNCTIONNAME> ... [SQL SECURITY {DEFINER | INVOKER}] AS ...

CREATE [OR ALTER] PROCEDURE <PROCEDURENAME> ... [SQL SECURITY {DEFINER | INVOKER}] AS ...

CREATE [OR ALTER] PACKAGE <PACKAGENAME> [SQL SECURITY {DEFINER | INVOKER}] AS ...
```

SQL SECURITY (Calculated columns)

```
connect 'localhost:/tmp/db.fdb' user sysdba password 'masterkey';
set term ^;
create function f() returns int
as
begin
    return 3;
end^
set term ;^

create table t (i integer, c computed by (i + f())) sql security definer;
insert into t values (2);
grant select on table t to user us;
grant execute on function f to user us;
commit;

connect 'localhost:/tmp/db.fdb' user us password 'pas';
select * from t;
```

SQL SECURITY (Functions)

```
connect 'localhost:/tmp/db.fdb' user sysdba password 'masterkey';
set term ^;
create function f (i integer) returns int sql security definer
as
begin
    insert into t values (:i);
    return i + 1;
end^
set term ;^
grant execute on function f to user us;
grant insert on table t to user us;
commit;

connect 'localhost:/tmp/db.fdb' user us password 'pas';
select f(3) from rdb$database;
```

SQL SECURITY (Stored procedures)

```
connect 'localhost:/tmp/db.fdb' user sysdba password 'masterkey';
set term ^;
create procedure p (i integer) sql security definer
as
begin
  insert into t values (:i);
end^
set term ;^
grant execute on procedure p to user us;
grant insert on table t to user us;
grant insert on table t to procedure p;
commit;

connect 'localhost:/tmp/db.fdb' user us password 'pas';
execute procedure p(1);
```

SQL SECURITY (Triggers)

```
connect 'localhost:/tmp/db.fdb' user sysdba password 'masterkey';
create table tr (i integer) sql security definer;
create table t (i integer);
set term ^;
create trigger tr_ins for tr after insert sql security definer
as
begin
    insert into t values (NEW.i);
end^
set term ;^
grant insert on table tr to user us;
grant insert on table t to user us;
commit;

connect 'localhost:/tmp/db.fdb' user us password 'pas';
insert into tr values(2);
```


SQL SECURITY (Packages)

```
connect 'localhost:/tmp/db.fdb' user sysdba password 'masterkey';
create table t (i integer);
set term ^;
create package pk sql security definer
as
begin
    function f(i integer) returns int;
end^
create package body pk
as
begin
    function f(i integer) returns int
    as
    begin
        insert into t values (:i);
        return i + 1;
    end
end^
set term ;^
grant execute on package pk to user us;
grant insert on table t to user us;
commit;

connect 'localhost:/tmp/db.fdb' user us password 'pas';
select pk.f(3) from rdb$database;
```

Thanks!

visit: www.red-soft.ru
roman.simakov@red-soft.ru