

# 오픈소스 데이터베이스 CUBRID의 개발 프로세스

---

2021-09-15

유형규 선임연구원

(주) 큐브리드



# 발표 내용

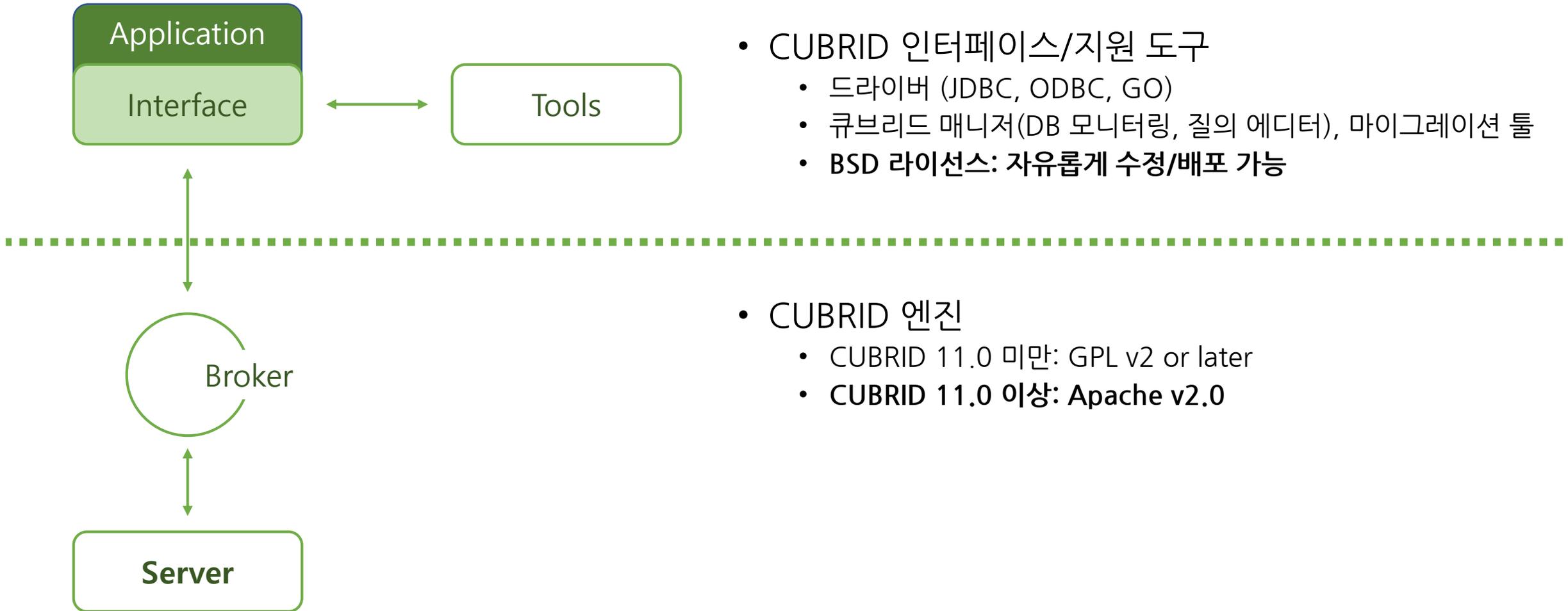
- CUBRID 소개
- 오픈소스?
- 오픈소스 거버넌스 모델
- 개발 프로세스
- 개발자 가이드 ([dev.cubrid.org](http://dev.cubrid.org))

# CUBRID 소개

- 국내 주도 엔터프라이즈 오픈소스 DBMS
- 최신 메이저 버전: v11.0
- 1년에 한번 마이너 버전 릴리즈, 4년에 한번 메이저 버전 릴리즈

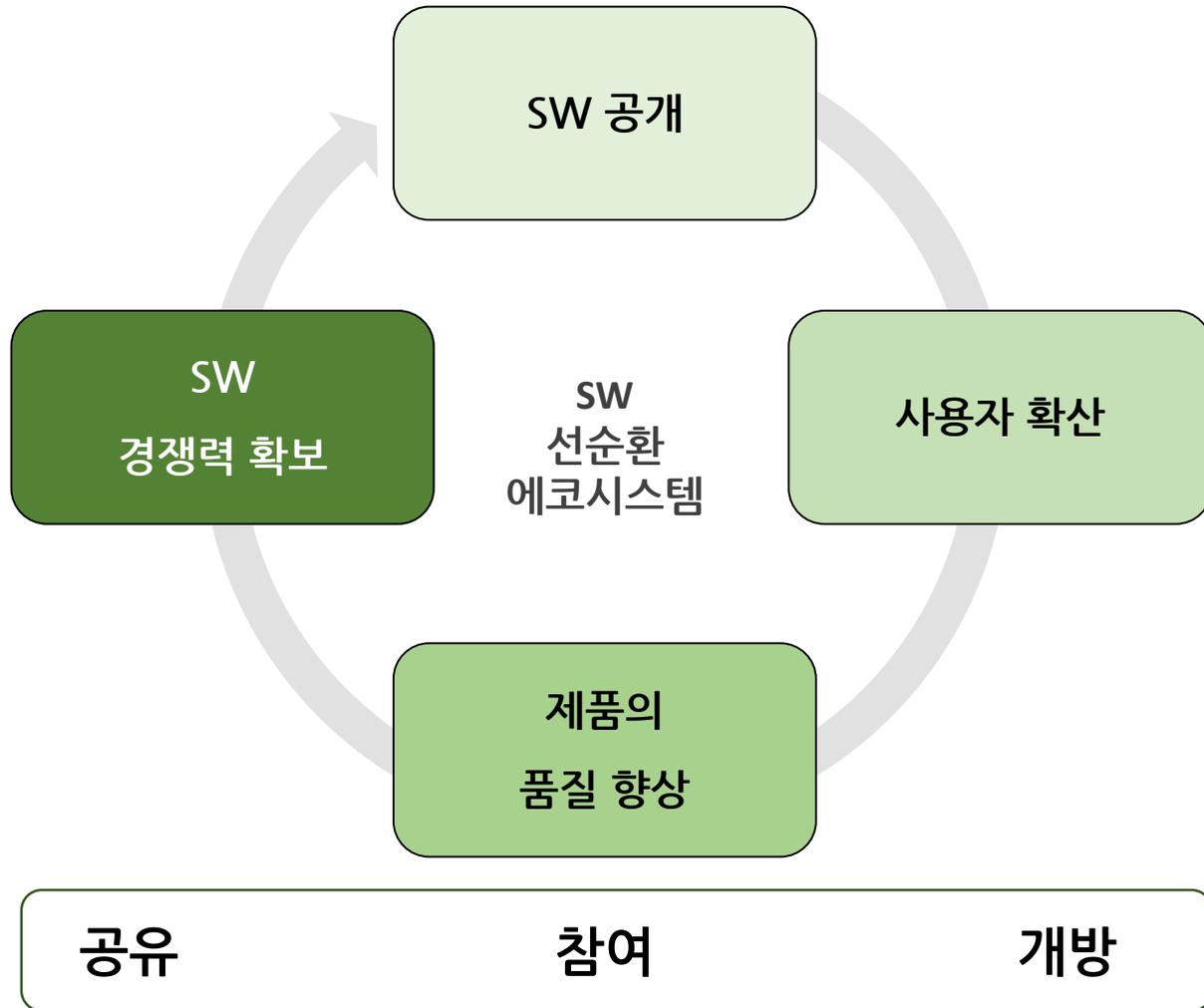


# CUBRID 오픈소스 라이선스



- CUBRID 인터페이스/지원 도구
  - 드라이버 (JDBC, ODBC, GO)
  - 큐브리드 매니저(DB 모니터링, 질의 에디터), 마이그레이션 툴
  - BSD 라이선스: 자유롭게 수정/배포 가능
- CUBRID 엔진
  - CUBRID 11.0 미만: GPL v2 or later
  - CUBRID 11.0 이상: Apache v2.0

# 오픈소스?



## 소스 코드만 공개한다고 오픈소스?

- 전체 중 일부의 코드만 공개
- (전체 코드가 공개되어 있더라도) 신규 기여자의 참여에 배타적
- 프로젝트 기여 방법을 비공개
- 프로젝트 진행 과정이 비공개

# 오픈소스 거버넌스 모델

오픈소스 프로젝트에서 “누가”, “무엇”을 그리고 “어떻게 해야 하는지”  
정의하는 규칙이나 관습, 프로세스

- “Do-ocracy”
- Founder-leader
- Self-appointing council or board
- Electoral
- Corporate-backed
- Foundation-backed

출처: <https://www.redhat.com/en/blog/understanding-open-source-governance-models>

# 오픈소스 거버넌스 모델 / “Do-ocracy”



## Do + Democracy

- 실제 업무를 수행하는 사람들의 의사 결정
- 서로 간의 커뮤니케이션/합의를 기반으로 작동
- 공식적이고 정교한 거버넌스 규칙은 만들지 않으려고 함
- 신규 기여자가 즉시 참여하기 어려울 수 있음

# 오픈소스 거버넌스 모델 / Founder-leader



리처드 스톨먼  
/ GNU 프로젝트



리누스 토르발스  
/ 리눅스

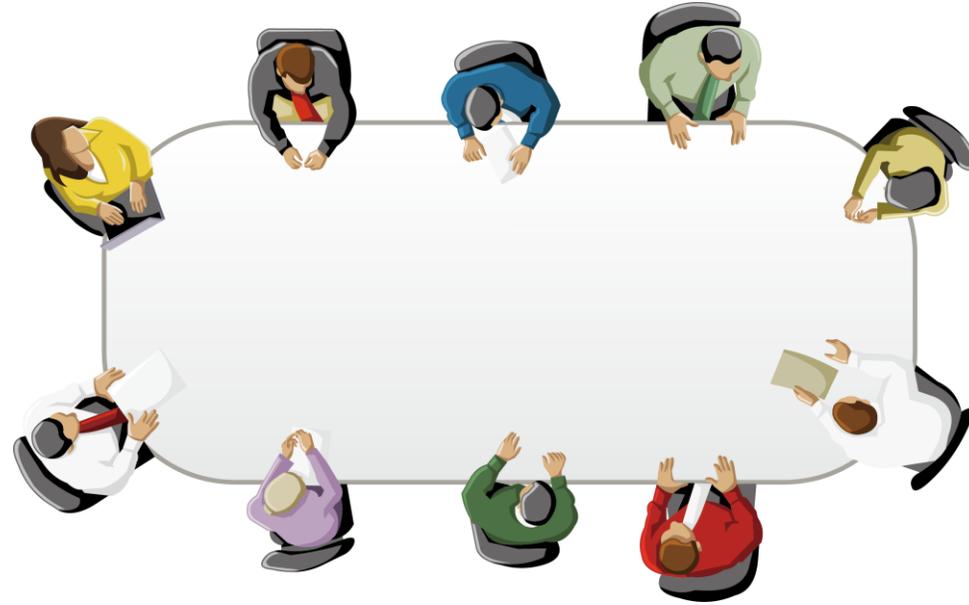


귀도 반 로섬  
/ 파이썬

- 프로젝트를 시작한 개인/그룹이 모든 권한을 제어
- 일부 프로젝트에서는 자비로운 종신독재자 (BDFL)이 존재
- 어느 규모 이상으로 성장하면 BDFL이 의사 결정 과정에서 병목 가능성
- 권한의 경계가 너무 명확

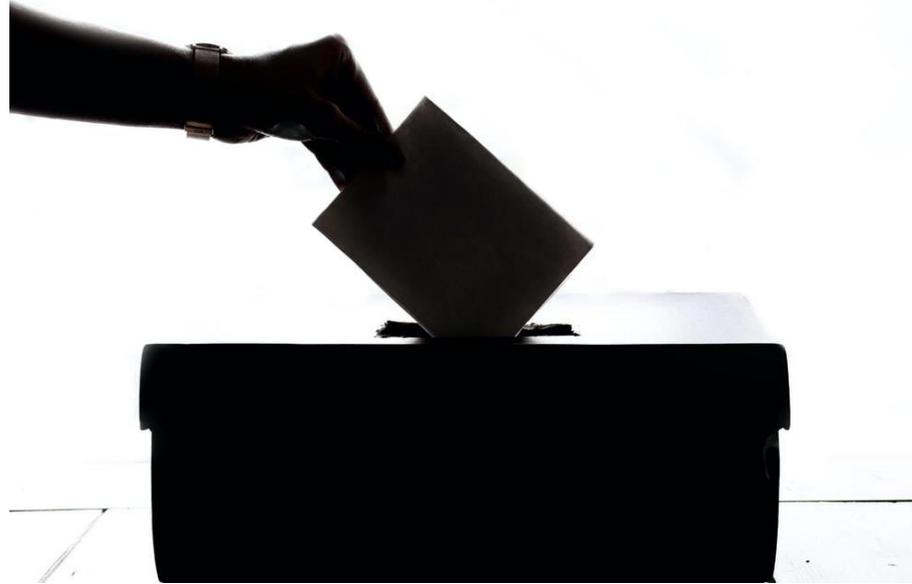
사진 출처: zetawiki

# 오픈소스 거버넌스 모델 / Self-appointing council or board



- 오픈소스 프로젝트의 다양한 측면을 관리하기 위해서 리더십 그룹을 지정  
ex) 기술 위원회, 인큐베이션 운영 위원회, 개발 프로세스 운영 위원회
- 관심 있는 작업에 주도적 참여를 하려면 리더십 그룹에 선택되어야 함  
리더십 활동에 대한 커뮤니티 참여에 방해 가능성

# 오픈소스 거버넌스 모델 / Electoral



- 다양한 프로젝트 역할을 수행할 후보자를 투표로 선출
- 프로젝트의 정책이나 절차를 승인하거나 업데이트 등 의사결정을 위한 투표
- 대규모의 오픈소스 프로젝트에 더 일반적
- 이를 위한 절차 수립/문서화/프로세스 필요, 시간이 많이 소요

그림 출처: <https://unsplash.com/@element5digital>

# 오픈소스 거버넌스 모델 / Corporate-backend



- 회사가 개발자나 사용자에게 다가가기 위한 방법으로 오픈소스 라이선스 조건에 따라 소프트웨어를 배포
- 작업의 빠른 의사 결정
- 소프트웨어의 개발, 플러그인 생태계 지원, 외부 개발자 커뮤니티 활성화를 위한 오버헤드를 피하고 회사의 리소스로 충족하기 위해 선택할 수 있음
  
- 오픈소스 프로젝트라고 주장하지만, 회사에 의해 완전히 통제되고 외부 기여자를 수락하지 않는 경우
- 신규 기여자 입장에서, 과연 오픈 소스 프로젝트가 맞는지 의문

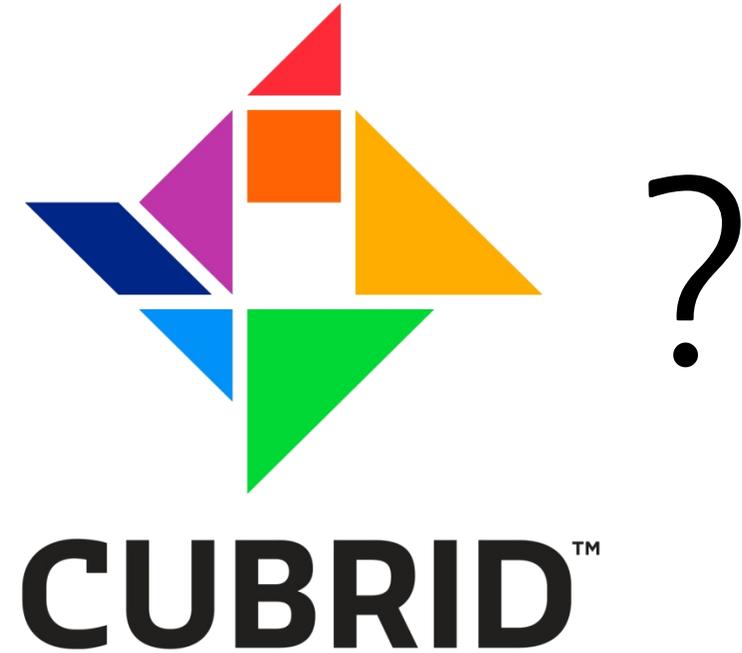
# 오픈소스 거버넌스 모델 / Foundation-backend



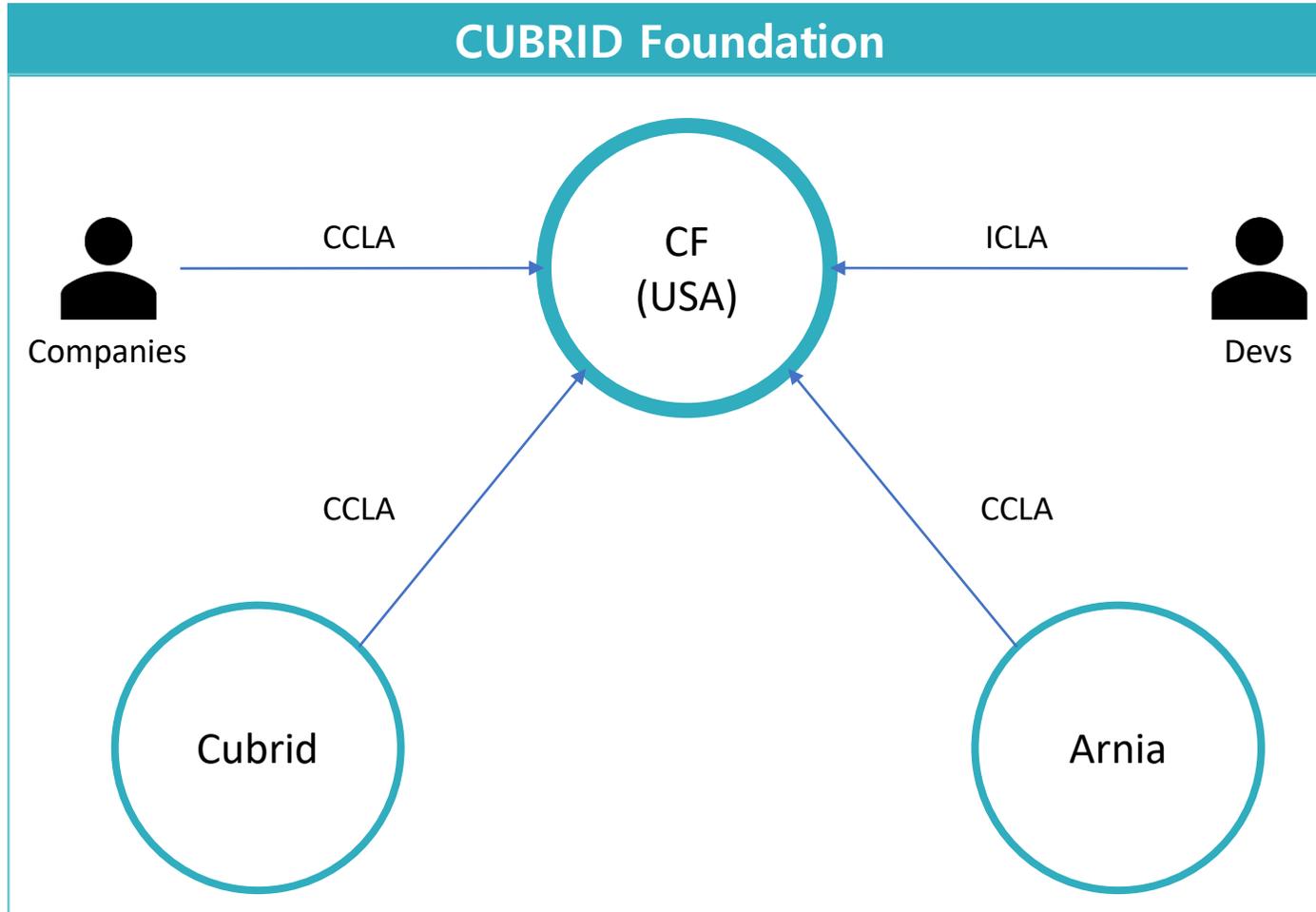
- 단일 프로젝트 참가자가 주요 프로젝트 리소스를 무조건 독점적으로 제어 불가능
- 1) 오픈 소스 프로젝트의 모든 측면을 관리하는 단일 거버넌스 구조
- 2) 상표나 이벤트와 같은 일부 문제를 관리하고, 기술적 의사결정은 프로젝트 리더가 관리 가능

# 오픈소스 거버넌스 모델

- “Do-ocracy”
- Founder-leader
- Self-appointing council or board
- Electoral
- Corporate-backed
- Foundation-backed

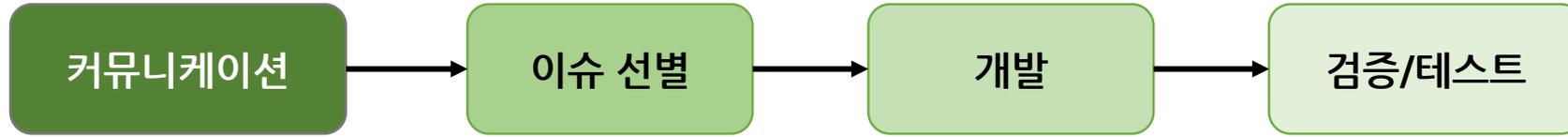


# CUBRID Foundation for Open Source Community



- 역할
  - 글로벌 오픈소스 생태계 구축 (관계사, 스폰서 확대 등)
  - CUBRID 차기 프로젝트를 조율/런칭
- 설립: 2020년 2월
- 위치: 워싱턴주 시애틀
- 대표: 김평철
- 대표 관계사
  - CUBRID Corp. / Korea
  - Arnia Corp. / Romania

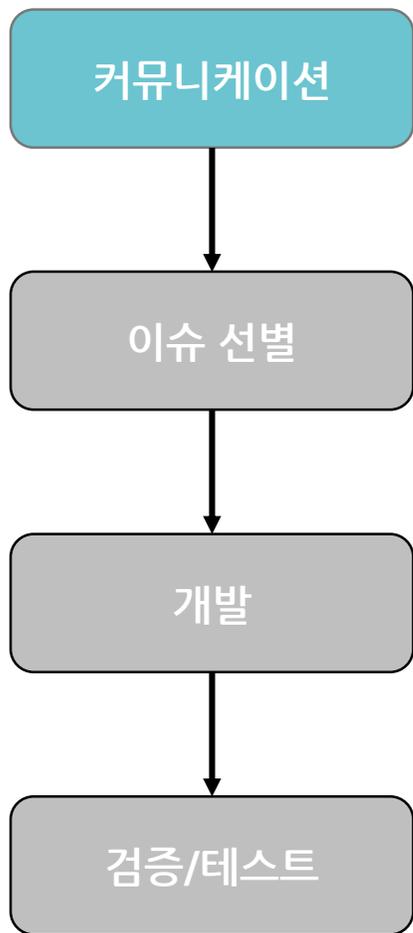
# CUBRID의 개발 프로세스



- CUBRID 프로젝트의 개발 프로세스에 오픈소스의 가치를 (공유/개방/참여) 담기 위한 노력
  - “CUBRID 프로젝트에 누구든 쉽고 안전하게 참여할 수 있고, 그 과정은 누구든 투명하고 볼 수 있으며 정보들이 충분히 잘 공유되어야 한다.“
- CUBRID 프로젝트를 리딩하는 개발조직과 신규 기여자가 모두 동일한 과정으로 참여
- 프로젝트 협업/자동화 도구를 기반으로 운영



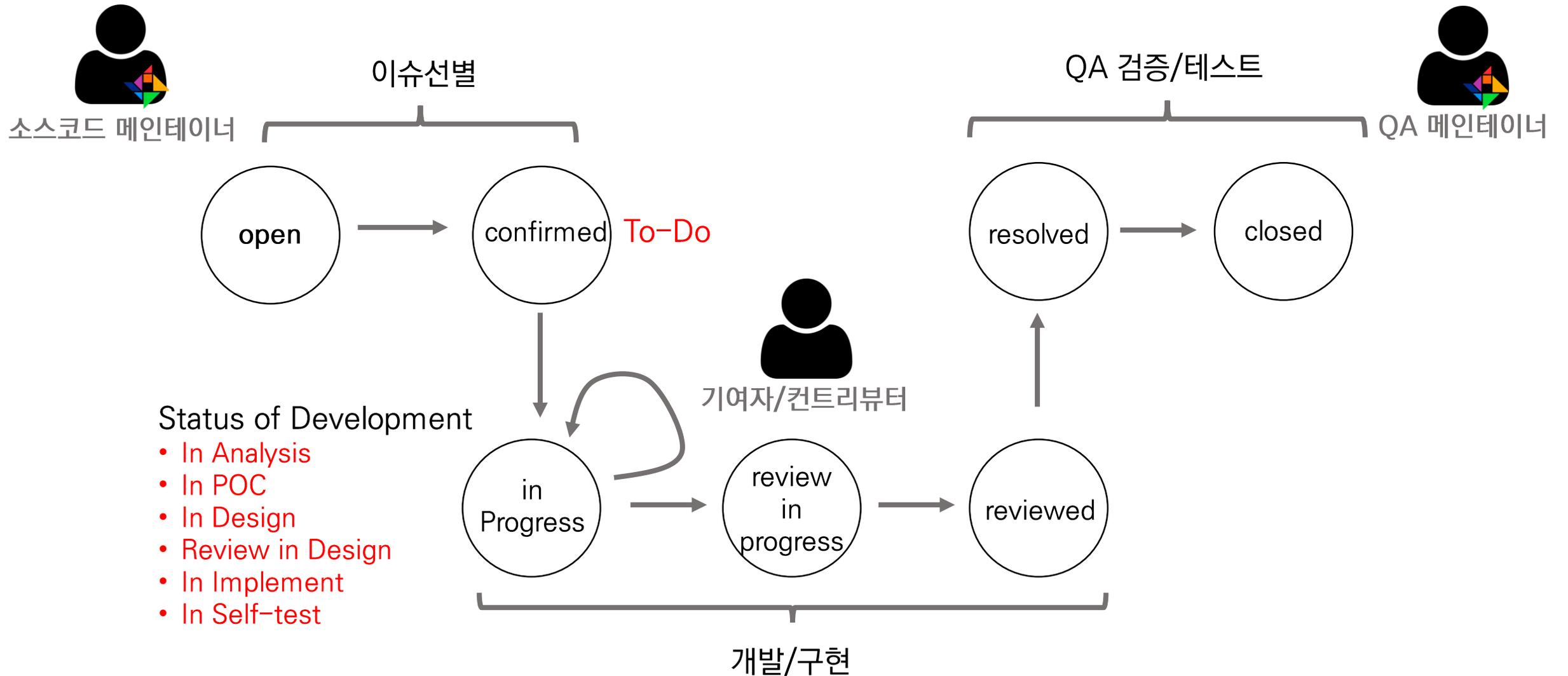
# 큐브리드의 개발 프로세스 / 커뮤니케이션



개발 참여를 위한 프로젝트, 기능 추가, 버그 수정에 대한 제안과 토론  
질문/답변, 정보 제공을 포함한 모든 오픈소스 커뮤니티 활동

- 버그/문제 토론, 질문, 정보 제공
  - (한글) cubrid.com, 자유게시판/Q&A
  - (영문) 레딧 CUBRID, <https://www.reddit.com/r/CUBRID/>
- 개발 토픽/버그 리포트
  - [jira.cubrid.org](https://jira.cubrid.org)
  - JIRA에 모든 의사결정과정/진행사항이 **공유**
  - 모든 프로젝트, 기능 추가, 버그 수정은 JIRA의 이슈 생성으로 시작

# 큐브리드의 개발 프로세스 / 커뮤니케이션



# 큐브리드의 개발 프로세스 / 커뮤니케이션

CUBRID / CBRD-23629  
Create utility for Java SP server

[Edit](#)
[Comment](#)
[Assign](#)
[More](#)
[Reopen Bug](#)
[Admin](#)

**Details**

Type:	↑ Improve Function/Performance	Status:	<b>CLOSED</b>
Priority:	✓ Minor	Resolution:	Fixed
Affects Version/s:	None	Fix Version/s:	None
Component/s:	CUBRID		
Labels:	<a href="#">cubrid.com</a> <a href="#">javasp</a>		
Planned Version(s):	damson		
Triage:	Approved		
QA Scenario:	Required		
DevStatus:	In Self-test		
Need Manual:	Yes		

**Description**

**Description**

When the java\_stored\_procedure system parameter is on, Java SP server is only able to start with the database server (cub\_server) together. So there was no way to server without restarting the cub\_server.

This work could be a preliminary task to add a system parameter to set JVM options for the Java server. perhaps a user wants to avoid restarting the database server to apply the options.

**Proposal**

I propose to develop a process that acts as a service utility of CUBRID. users can manage a utility by the following command, which performs java stored procedure query.

**Usage example**

```
usage: cubrid javasp <command> [args]
```

Available command:

```
start    [database-name]
stop     [database-name]
restart  [database-name]
status   [database-name]
```

*cubrid javasp status* will show the following information

- The database name
- The process id
- The port number for which communicate with CAS
- Applied JVM options

**Architecture Overview**

The followings are added newly in the figure above:

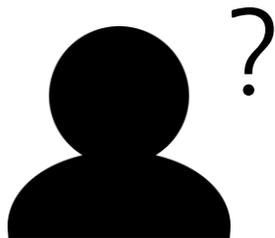
- **cub\_javasp**: It has the main routine for javasp server process. it processes commands given by cubrid utility.
- **\$CUBRID/var/javasp/javasp\_<db\_name>.info**: a file to store pid and port number of running javasp server. When javasp server start, PID and port number are written in this file. From other commands, cubrid utility reads this file and try to connect with written information to the running javasp server and check the status of the server.
- **\$CUBRID/log/<db\_name>\_java.err**: error message is written when error occurs in process of startup JVM.
- **Able to be managed by 'cubrid service' command**: javasp server is a service utility so that users can register it in the service section in the conf file (\$CUBRID/conf/cubrid.conf) and manage this utility by 'cubrid service' command.

In Standalone mode (SA\_MODE), it operates as a thread in cub\_server as before. In CS\_MODE, users can start and manage the javasp server with the cubrid utility or the cub\_javasp utility.

# 큐브리드의 개발 프로세스 / 커뮤니케이션

## 신규 기여자의 참여를 저해하는 요인

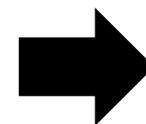
- 복잡한 절차
- 가이드라인 부족



## 예시) JIRA의 경우

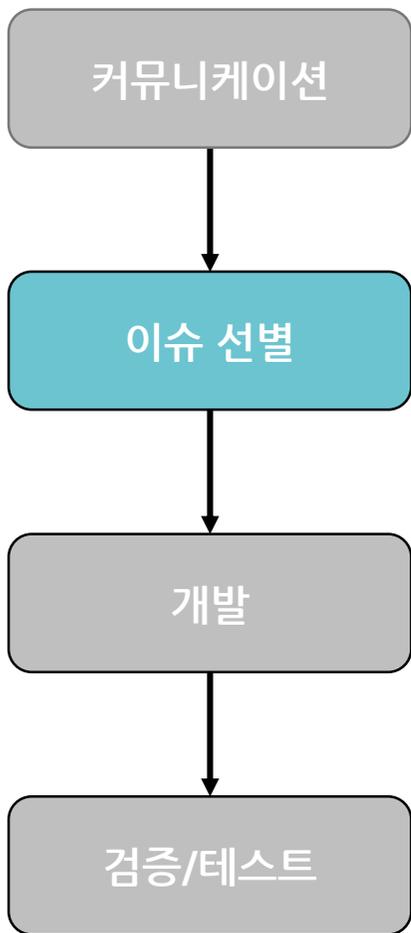
- 언제 이슈를 열어도 되는지 모르겠음
- 이슈를 만들려고 하니 입력해야 할 항목들이 너무 많이 보여서 언제 어떤 항목을 작성해야 할 지 모르겠음
- 어떤 내용을 작성해야 할지 모르겠음

- 1) 개발 프로세스 이해를 위한 가이드라인 문서 작성
- 2) 이슈의 종류 (버그, 기능개선 등) 에 따라 작성할 내용 가이드



**개발자 가이드 필요!**

# 큐브리드의 개발 프로세스 / 이슈 선별

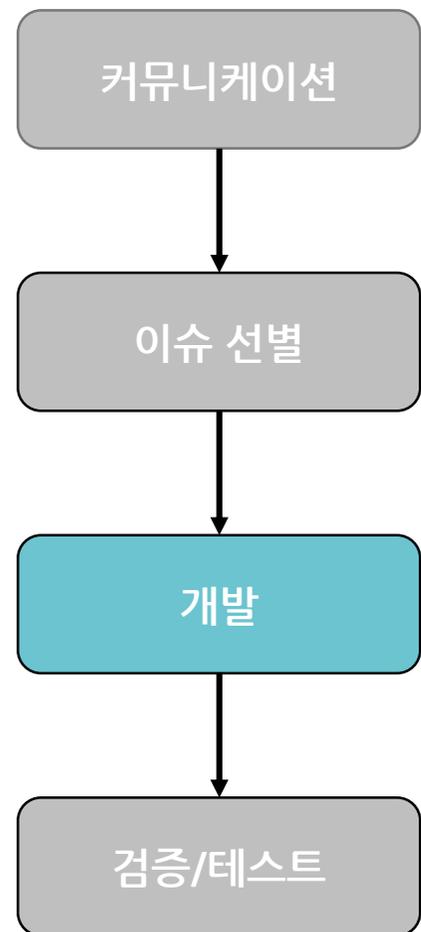


- CUBRID Foundation: CUBRID의 전체 프로젝트 단위 로드맵/비전 컨트롤
- CUBRID 서브 프로젝트 (엔진, 드라이버, QA 도구, 여러가지 툴)
  - 실제 해당 프로젝트를 담당하는 개발조직에서 오픈소스 메인테이너 역할

Project	Category	Key
CUBRID	Engine	CBRD
CUBRID Affiliates		AFFILIATES
CUBRID APIs	Driver	APIS
CUBRID Apps		PORTING
CUBRID Cluster		CLUSTER
CUBRID Engine (obsolete)		ENGINE
CUBRID LETS		LETS
CUBRID Manuals		CUBRIDMAN
CUBRID QA	QA System	CUBRIDQA
CUBRID Tools	Tool	TOOLS
Cubrid.org		CUBRIDORG



# 큐브리드의 개발 프로세스 / 개발



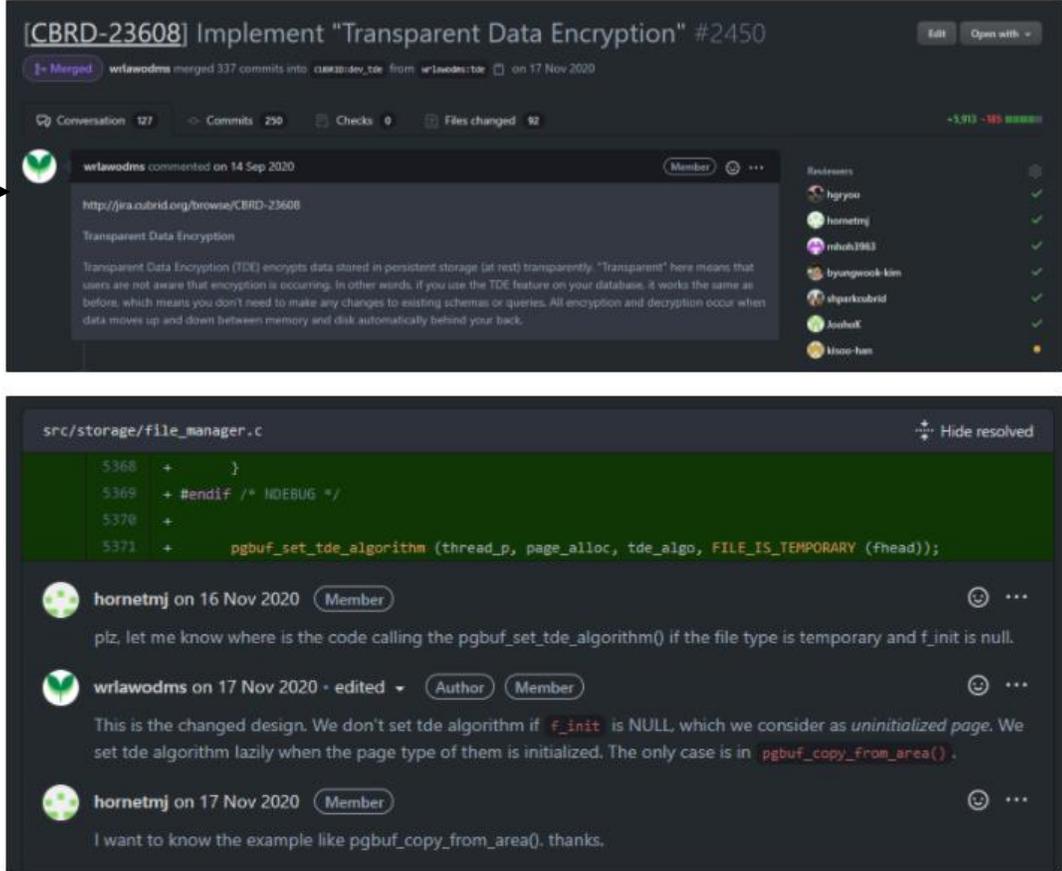
Github의 Pull Request 기능을 이용해 코드 리뷰 수행

**코드리뷰!**

구현결과



**CUBRID™**



The screenshot shows a GitHub Pull Request titled "[CBRD-23608] Implement 'Transparent Data Encryption' #2450". It includes a commit message, a code diff for `src/storage/file_manager.c`, and several review comments. One comment from `honetmj` asks for clarification on the `pgbuf_set_tde_algorithm()` function call. Another comment from `wrlawodms` explains the design change regarding `f_init` being NULL. A final comment from `honetmj` asks for an example of `pgbuf_copy_from_area()`.

# 큐브리드의 개발 프로세스 / 개발

개발 프로세스에서 가장 중요한 목표 중 하나는 **코드 리뷰**

- 더 좋은 코드 수준, 결함 발견, 지식 공유, 책임감 등의 장점
- 어느 수준까지 리뷰해야 하는지에 대한 고민
- 무조건 높은 수준을 요구할수록 코드 리뷰 완료 시간이 길어짐
- 신규 기여자가 더 이상의 프로세스 진행을 포기할 수 있음
- 리뷰어의 리소스도 한정적임



효과적/효율적인 코드 리뷰를 유도하기 위한 고민과 개선이 필요

# 큐브리드의 개발 프로세스 / 개발

## 자동화 도구 도입

- 지루하고, 컴퓨터가 더 잘 할 수 있는 부분에는 자동화 도구를 활용
- 코드 스타일, 라이선스 주석 확인, 정적 분석
- 각 환경 별 (CentOS, Ubuntu, Windows) 빌드 후 성공 여부 확인
- 많은 SQL 테스트 케이스를 돌려보고 기능에 문제가 없는지 검증

```
Check code style
20 else if (reply_code == ER_CDC_ADJUSTED_LSA)
21 - {
22 - ptr = or_unpack_log_lsa (ptr, lsa);
23 - or_unpack_int64 (ptr, timestamp);
24 + {
25 + ptr = or_unpack_log_lsa (ptr, lsa);
26 + or_unpack_int64 (ptr, timestamp);
27
28 if (recv_data != NULL && recv_data != reply)
29 {
30 free_and_init (recv_data);
31 }
32
33 - return CUBRID_LOG_SUCCESS_WITH_ADJUSTED_LSA;
34 - }
35 + return CUBRID_LOG_SUCCESS_WITH_ADJUSTED_LSA;
36 + }
```

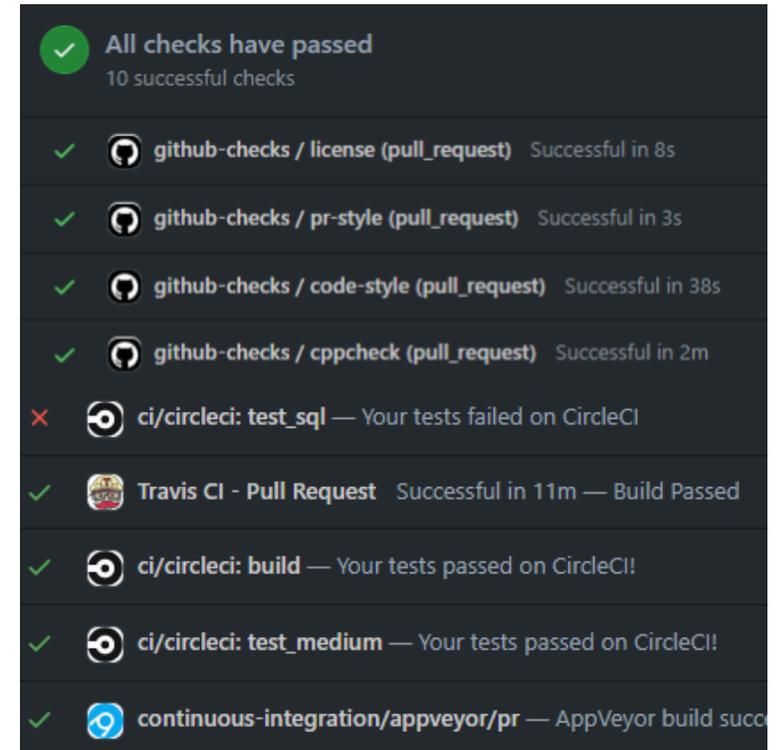
코드 스타일 확인



10 tests failed out of 15701

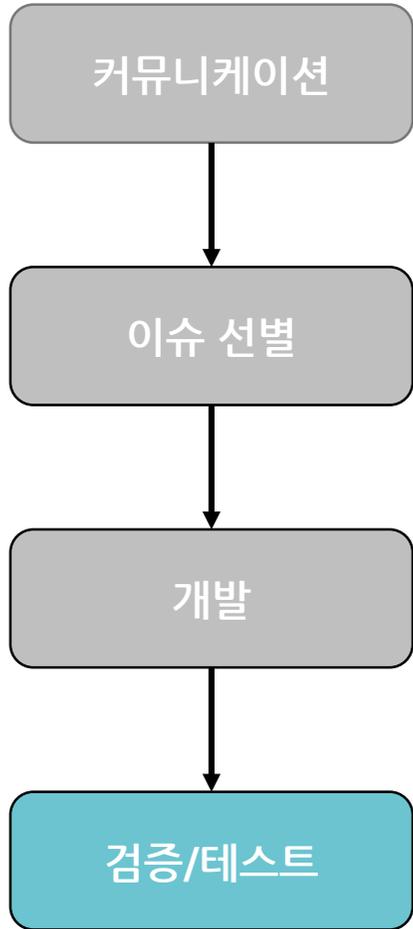
- [sql/\\_31\\_cherry/issue\\_22161\\_CTE\\_extensions/\\_02\\_delete/cases/subquery\\_001.sql](#)
- [sql/\\_31\\_cherry/issue\\_22161\\_CTE\\_extensions/\\_05\\_create/cases/cbrd\\_22582.sql](#)
- [sql/\\_31\\_cherry/issue\\_22161\\_CTE\\_extensions/\\_05\\_create/cases/table\\_of\\_db\\_partition.sql](#)
- [sql/\\_13\\_issues/\\_14\\_2h/cases/bug\\_bts\\_13764.sql](#)
- [sql/\\_13\\_issues/\\_15\\_1h/cases/bug\\_bts\\_7547\\_2.sql](#)
- [sql/\\_13\\_issues/\\_20\\_2h/cases/cbrd\\_23665.sql](#)
- [sql/\\_13\\_issues/\\_20\\_2h/cases/cbrd\\_23749.sql](#)
- [sql/\\_13\\_issues/\\_21\\_1h/cases/cbrd\\_23877.sql](#)
- [sql/\\_13\\_issues/\\_21\\_2h/cases/cbrd\\_24003.sql](#)
- [sql/\\_23\\_apricot\\_qa/\\_02\\_performance/\\_01\\_filtered\\_index/cases/filtered\\_index\\_data\\_type\\_basic\\_view.sql](#)

SQL 테스트



그 외 여러가지 자동화 도구

# 큐브리드의 개발 프로세스

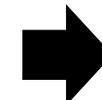


**Jenkins**

cubrid-testtools 프로젝트의 테스트 도구가 Jenkins에서 실행  
(<https://github.com/cubrid/cubrid-testtools>)

linux

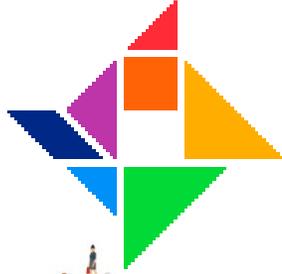
Bits	Category	Total	Testing	Success	Fail		Rate		Elapse Time	Test Date	Unstable Case			
					Total	New	Test Rate	Fail Rate			Total	New	Trend	Verified Rate
64bits	sql	15698	15698	15698	0	0	100%	0%	3012	2021-02-25 00:22	0	0	0	100%
	sql_debug	15698	15698	15698	0	0	100%	0%	10532	2021-02-25 00:23	0	0	0	100%
	medium	970	970	970	0	0	100%	0%	232	2021-02-25 00:21	0	0	0	100%
	medium_debug	970	970	970	0	0	100%	0%	860	2021-02-25 00:23	0	0	0	100%
	sql_by_cci	15698	15698	15697	1	1	100%	1.00%	2921	2021-02-25 01:10	1	1	+1	100%
	shell	2809	2781	2777	4	1	100%	0.14%	13777	2021-02-25 11:30	0	0	0	100%
	shell_debug	2809	2781	2772	9	6	100%	0.32%	26321	2021-02-25 07:39	6	6	+6	100%
	shell_heavy	100	100	99	1	0	100%	1.00%	24612	2021-02-26 15:10	0	0	0	100%
	shell_long	71	71	71	0	0	100%	0%	60658	2021-02-26 08:18	0	0	0	100%
	cci	321	321	321	0	0	100%	0%	15023	2021-02-25 09:28	0	0	0	100%
	cci_debug	321	321	321	0	0	100%	0%	17853	2021-02-25 05:17	0	0	0	100%
	ha_shell	323	323	320	3	1	100%	0.93%	14109	2021-02-25 04:58	1	1	+1	100%
	ha_repl	15698	15504	15503	1	1	100%	0.01%	32089	2021-02-26 01:52	1	1	+1	100%
	ha_repl_debug	15698	15504	15504	0	0	100%	0%	59437	2021-02-25 16:52	0	0	0	100%
	isolation	6770	6752	6752	0	0	100%	0%	1981	2021-02-25 05:31	0	0	0	100%
	isolation_debug	6770	6752	6752	0	0	100%	0%	2539	2021-02-25 01:02	0	0	0	100%



개발자

cubrid-testcases 프로젝트의 테스트 케이스  
(<https://github.com/cubrid/cubrid-testcases>)

# 개발 프로세스 개선 고민과 개발자 가이드 작성



- 큐브리드에 참여하는 개발조직이 커짐
- 개발 프로세스에 대한 의견 일치 과정이 필요
- 신규 기여자를 위한 큰 그림의 가이드가 없음
- 큐브리드에 기여하기 위한 정보가 흩어져 있음

## 개발자 가이드를 작성하자

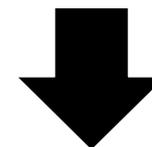


Image: Designed by iconicbestiary / Freepik

# 개발자 가이드 작성 시 고려사항



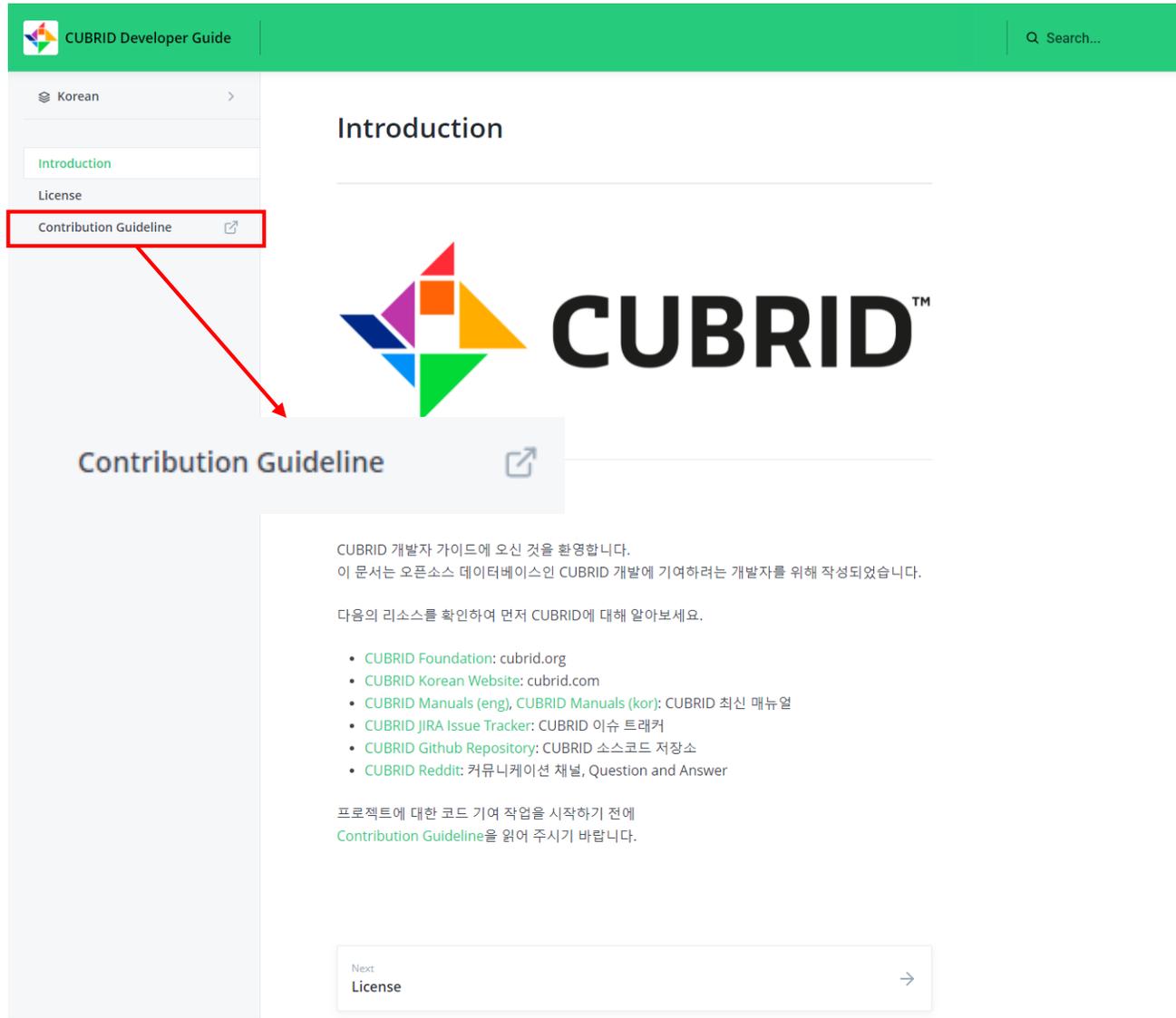
- 보기 힘든 문서는 읽지 않는다
- 쉽게 최신 버전을 업데이트하고 볼 수 있어야 한다
- 버전 관리가 쉬워야 한다
- 완벽한 개발 프로세스도 가이드 문서도 없다
- 쉽게 편집할 수 있어야 한다



**GitBook**

+ Open Source Community Plan

# 개발자 가이드 - dev.cubrid.org



CUBRID Developer Guide

Korean

Introduction

License

Contribution Guideline

## Introduction

### Contribution Guideline

CUBRID 개발자 가이드에 오신 것을 환영합니다.  
이 문서는 오픈소스 데이터베이스인 CUBRID 개발에 기여하려는 개발자를 위해 작성되었습니다.

다음의 리소스를 확인하여 먼저 CUBRID에 대해 알아보세요.

- CUBRID Foundation: cubrid.org
- CUBRID Korean Website: cubrid.com
- CUBRID Manuals (eng), CUBRID Manuals (kor): CUBRID 최신 매뉴얼
- CUBRID JIRA Issue Tracker: CUBRID 이슈 트래커
- CUBRID Github Repository: CUBRID 소스코드 저장소
- CUBRID Reddit: 커뮤니케이션 채널, Question and Answer

프로젝트에 대한 코드 기여 작업을 시작하기 전에  
[Contribution Guideline](#)을 읽어 주시기 바랍니다.

Next  
License →

- 개발 프로세스 전반에 대한 설명
- 각 단계 별 기여 가이드
  - JIRA 이슈 작성과 필수 내용
  - 이슈 상태 별 절차 정의
  - 코드 리뷰
  - 자동화 도구 설명
  - QA 리포트 확인과 테스트 실패 시 절차
  - 매뉴얼 작성 가이드
  - 신규 기여자 가이드
- 한글, 영문 버전 지원

# 마치며...

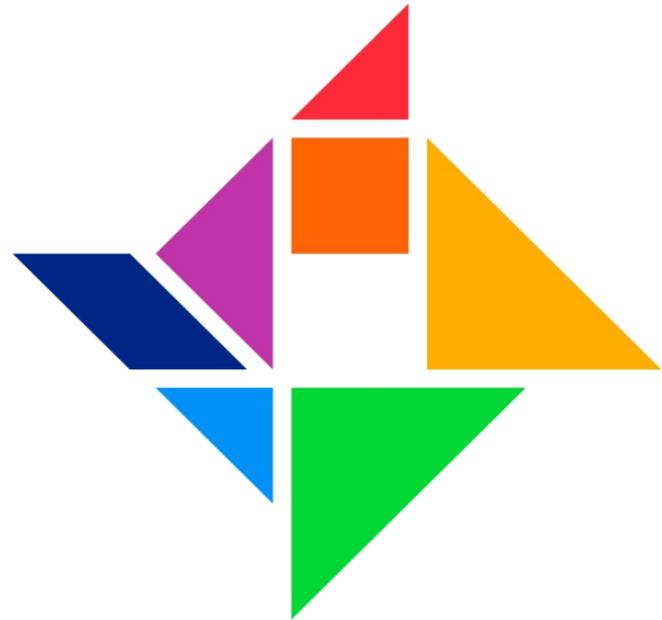


- 오픈소스 커뮤니티는 계속 발전하면서 변합니다
- 더 나은 퀄리티의 결과와 높은 생산성의 균형을 위한 고민
- 신규 기여자의 성공적인 참여를 위한 고민

## What's Next?

CUBRID 개발 시 프로젝트를 이해를 더 도와줄 수 있는 내용

- 더 자세한 빌드 방법
- 디버깅/성능 측정 가이드
- 소스코드의 모듈 별 자세한 분석 자료
- 좀 더 기술적인 내용을 커뮤니티에 공유



**CUBRID™**

감사합니다