

gRPConf

INDIA 2026

gRPConf INDIA 2026

Production-Grade gRPC APIs: Contract Governance, Security Validation, Zero-Breaking Deployments

SPEAKERS

Dillibabu Sampath

Lead Software Engineer, Wells Fargo

Malik Chandra Pandian T

Senior Engineer, Cloud Engineering, Brillio

01 A build can pass. An API can break.

Question: who validates the contract that existing consumers rely on?

APPLICATION BUILD PASS

CONSUMER COMPATIBILITY UNKNOWN

Build success ≠ safe API release

02 One small .proto change

A field number is part of the Protobuf wire contract.

BEFORE

```
message User {
  string username = 2;
}
```

AFTER

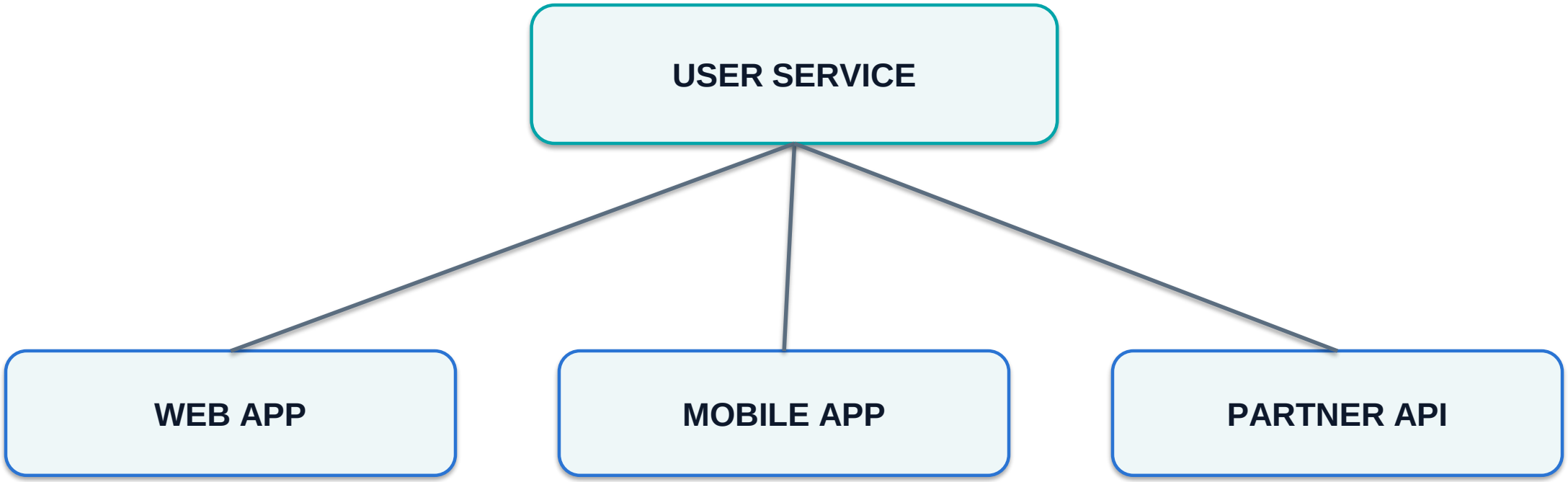
```
message User {
  string name = 3;
}
```

Rename: may affect source / JSON consumers.

Field number 2 → 3: breaks the previous wire identity.

03 One service. Multiple consumers.

A shared contract creates a release dependency across teams.



Existing clients cannot all be upgraded atomically.

04 What traditional CI/CD misses

gRPConf INDIA 2026

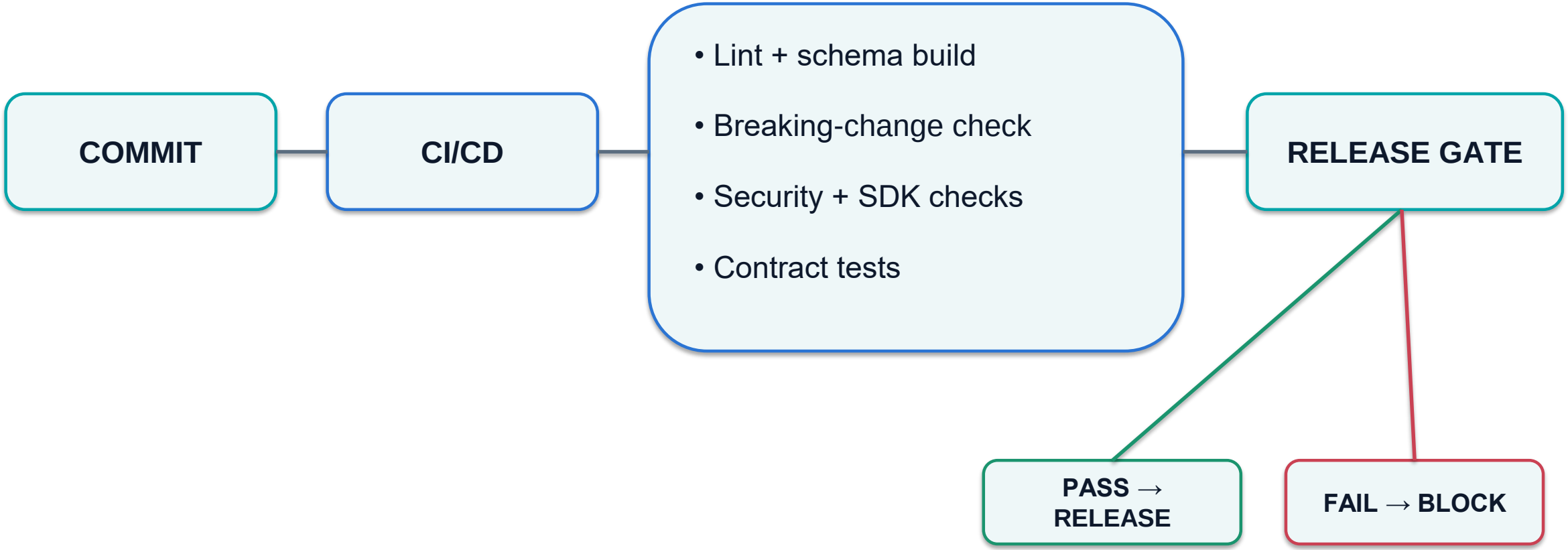
Application checks and API compatibility checks answer different questions.

- ✓ Compile
- ✓ Unit tests
- ✓ Build image
- ✓ Deploy health

- ! Schema compatibility
- ! Consumer SDK compatibility
- ! Security policies
- ! Behavioral compatibility

05 Shift API validation left

Independent validation checks feed one mandatory release decision.



06 Five layers. Five failure categories.

Every layer answers a different production-readiness question.

01	**Buf lint**	Conventions & schema quality
02	**Buf breaking**	Backward compatibility
03	**Protovalidate + policy tests**	Constraints & security controls
04	**Buf generate + Go**	Generated SDK compiles
05	**Go gRPC contract tests**	Service and consumer behavior

07 Compatibility is a policy choice

Choose a rule based on the guarantees you promise consumers.

WIRE	Binary wire-level compatibility
WIRE_JSON	Wire and JSON-facing compatibility
FILE / PACKAGE	Stricter generated API compatibility

Compare with a trusted released baseline, not an arbitrary commit.

08 The checks are executable

Representative Buf commands for the User Service repository.

```
buf build
buf lint
buf breaking --against .git#branch=main
buf generate
go test ./...
```

SDK builds and security tests must be separately wired into CI.

09 Contract tests protect behavior

Schema checks cannot prove runtime semantics or authorization.

SCHEMA	Field numbers, names and types
SECURITY	Input constraints + auth policy tests
BEHAVIOR	Old client requests and response semantics

10 The release gate is the product

Fail closed on mandatory validation failures.



Intentional breaks require versioning, migration and explicit approval.

LIVE DEMO | 5 minutes

One User Service. One breaking change. One release decision.

1 Show released baseline and existing client

2 Change username = 2 to name = 3

3 Run go build ./... → build can pass

4 Run buf breaking against baseline → FAIL

5 Show release gate → BLOCKED

KEY TAKEAWAYS + Q&A

gRPConf INDIA 2026

Three things to remember

Build ≠ compatibility

A green build can still break existing clients.

Governance belongs in CI/CD

Validate contracts, security and consumer behavior.

Gate the release

Block promotion when mandatory checks fail.

Production-Grade gRPC APIs: Contract Governance, Security Validation, Zero-Breaking Deployments

Questions? Let’s discuss.

Scan to share feedback

REFERENCES

Official documentation and implementation guidance

- Buf CLI / lint / breaking: https://buf.build/docs/
- Protovalidate: https://buf.build/docs/protovalidate/overview/
- gRPC: https://grpc.io/docs/
- Protocol Buffers: https://protobuf.dev/programming-guides/proto3/
- grpcurl: https://github.com/fullstorydev/grpcurl