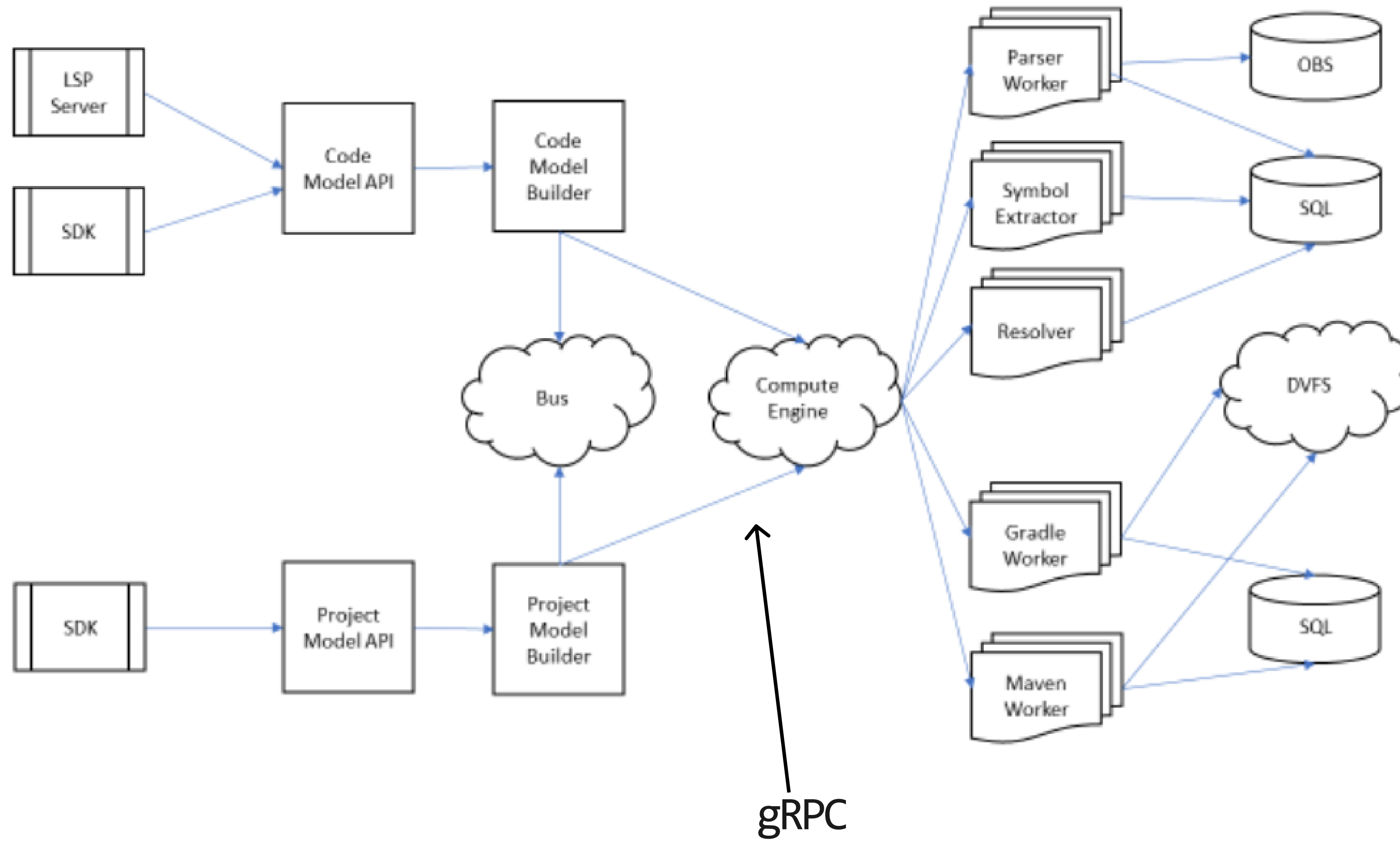


High-level code generator for java-protobufs

SERGEY ANCHIPOLEVSKIY, MENTOR (SUDU TEAM)

DMITRY DUZHINSKY, INTERN

The Sudu project background



Protobuf model lifecycle

Unsuitable to use
in business logic

proto

click build →

Google
-generated



manually written



Business logic

Error-prone

Takes a lot of time

```
message GrpcBatchInfo {  
  GrpcBatchId batch = 1;  
  map<string, string> commonParameters = 2;  
  repeated GrpcTaskInfo task = 3;  
  string clientTag = 4;  
  string taskType = 5;  
}
```

Dozens of models

```
public record BatchInfo(  
    @NotNull String uuid,  
    @NotNull Map<String, String> commonParameters,  
    @NotNull List<TaskInfo> tasks,  
    @NotNull String clientTag,  
    @NotNull String taskType  
) {  
  
    @NotNull  
    public static BatchInfo fromGrpc(@NotNull GrpcBatchInfo proto) {...}  
  
    @NotNull  
    public GrpcBatchInfo toGrpc() {...}  
}
```

Problem statement

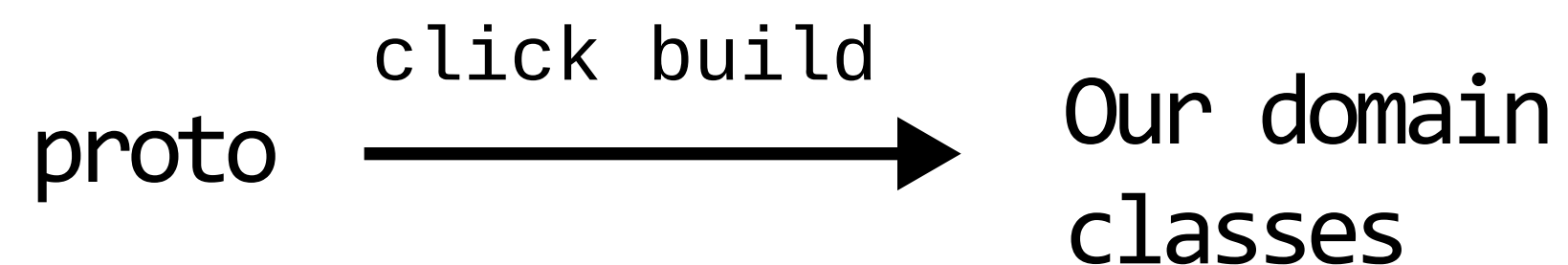
What if we created a tool that relieves a developer from this?

```
public GrpcBatchInfo toGrpc() {
    GrpcBatchInfo.Builder builder = GrpcBatchInfo.newBuilder();
    builder.setBatch(GrpcBatchId.newBuilder().setUuid(uuid).build());
    builder.putAllCommonParameters(commonParameters);
    builder.addAllTask(tasks.stream()
        .map(i -> i.toGrpc())
        .collect(Collectors.toList()));
    builder.setClientTag(clientTag);
    builder.setTaskType(taskType);
    return builder.build();
}
```

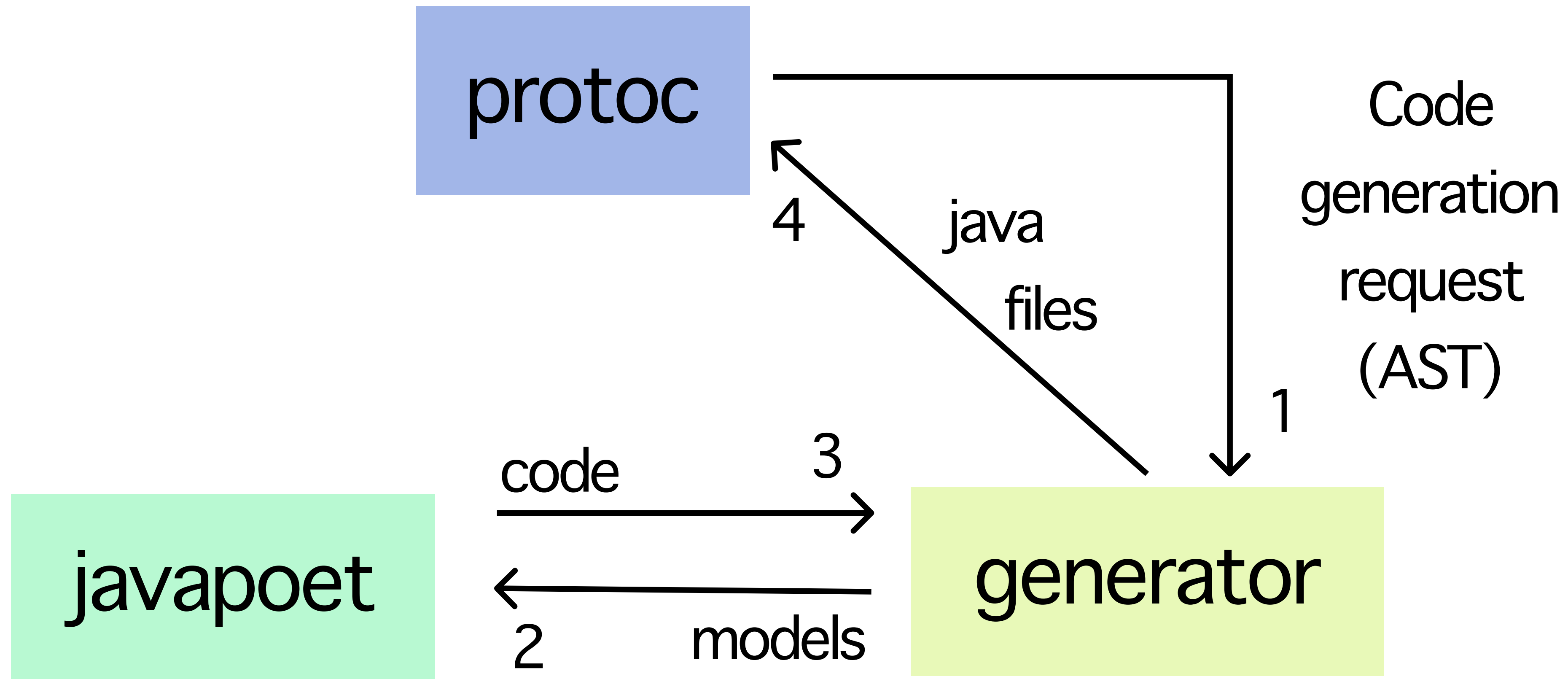
Now



With such tool



Solution architecture



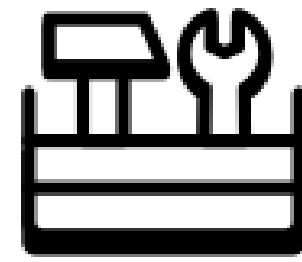
Results

- The tool has been already integrated into the project
- Console and compute-engine have started using it

 Merged

Future plans

- Generate asynchronous clients
- Provide more features



High-level code generator for java-protobufs

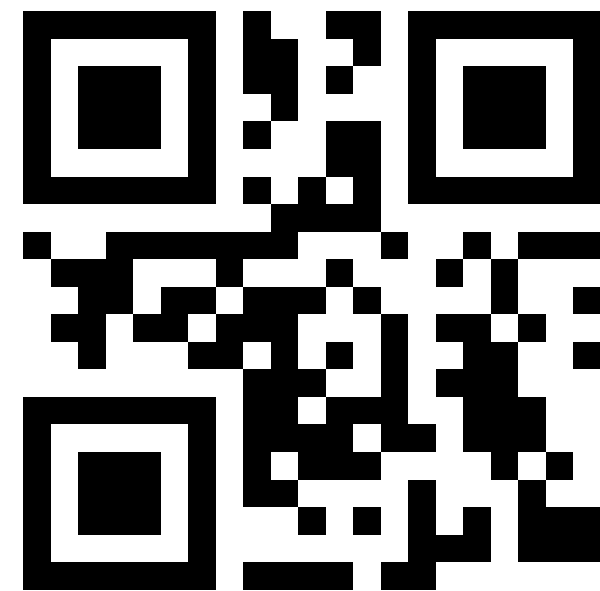
SERGEY ANCHIPOLEVSKIY, MENTOR (SUDU TEAM)

DMITRY DUZHINSKY, INTERN

Contacts:

dduzhinsky@ya.ru

t.me/duzhinsky



Repository:

github - [SuduIDE/sudu](#)

module [protogen](#)