7 years of talking to people

it's more important to give a talk than to do it well

Jakub Daliga

part 0: who am I

Golang Warsaw #12 (29.06.2017)
Jakub Daliga - GraphQL w Go (PL)

notable experience

~9 years of professional programming

6 companies

smallest: 3 people

largest: 100k+ people

working with people from 3 continents

5 talks given on Golang Warsaw

(this is 6th)

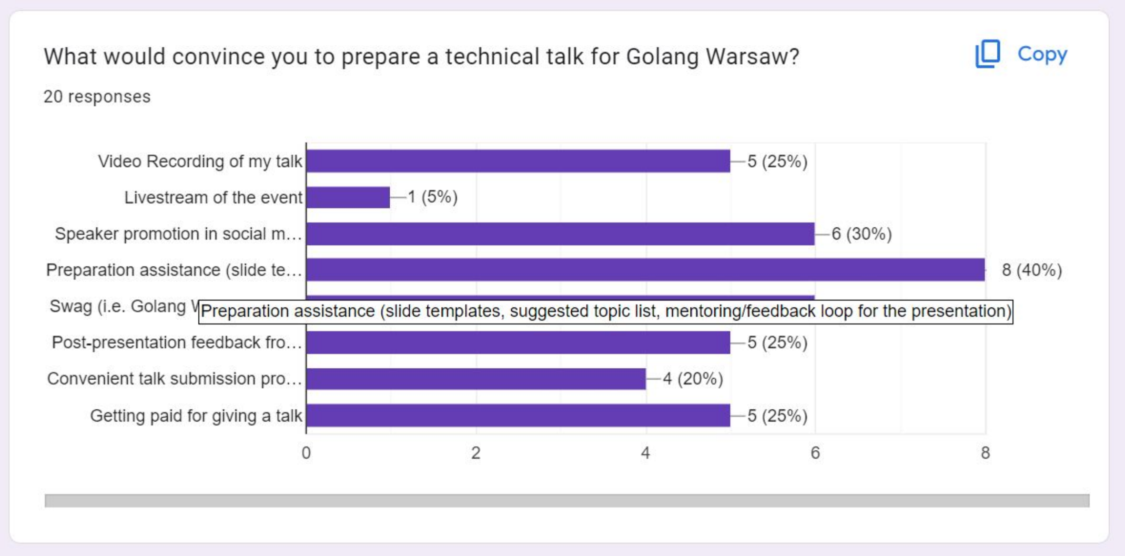
co-organizing the event

this talk

part 0: who am I

part 1: philosophical outlook

part 2: practical TODO list

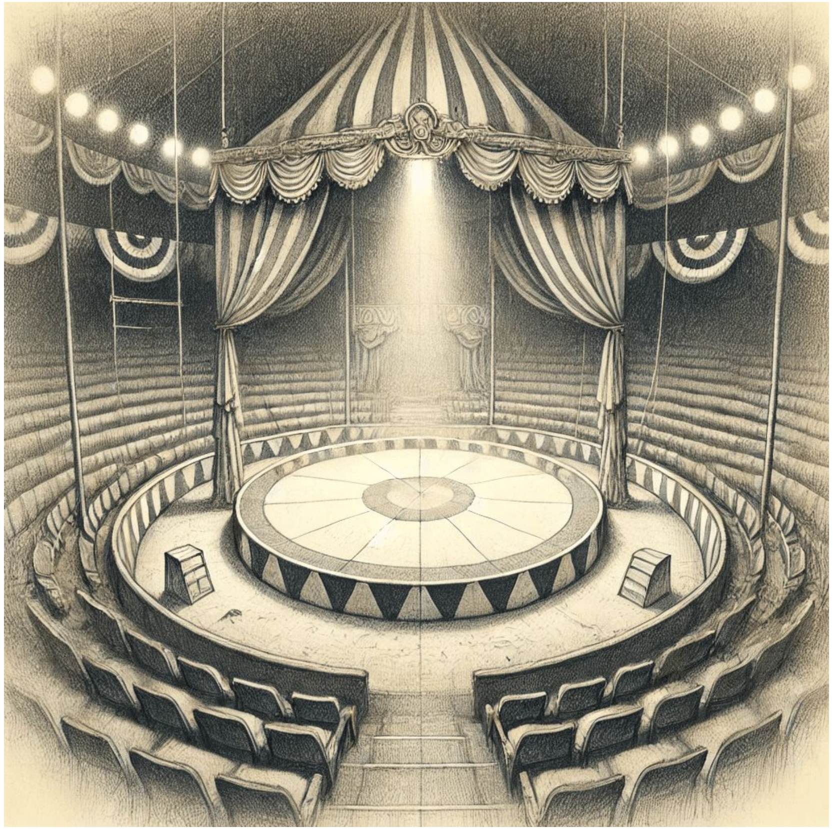


What would convince you to prepare a technical talk for Golang Warsaw?

20 responses

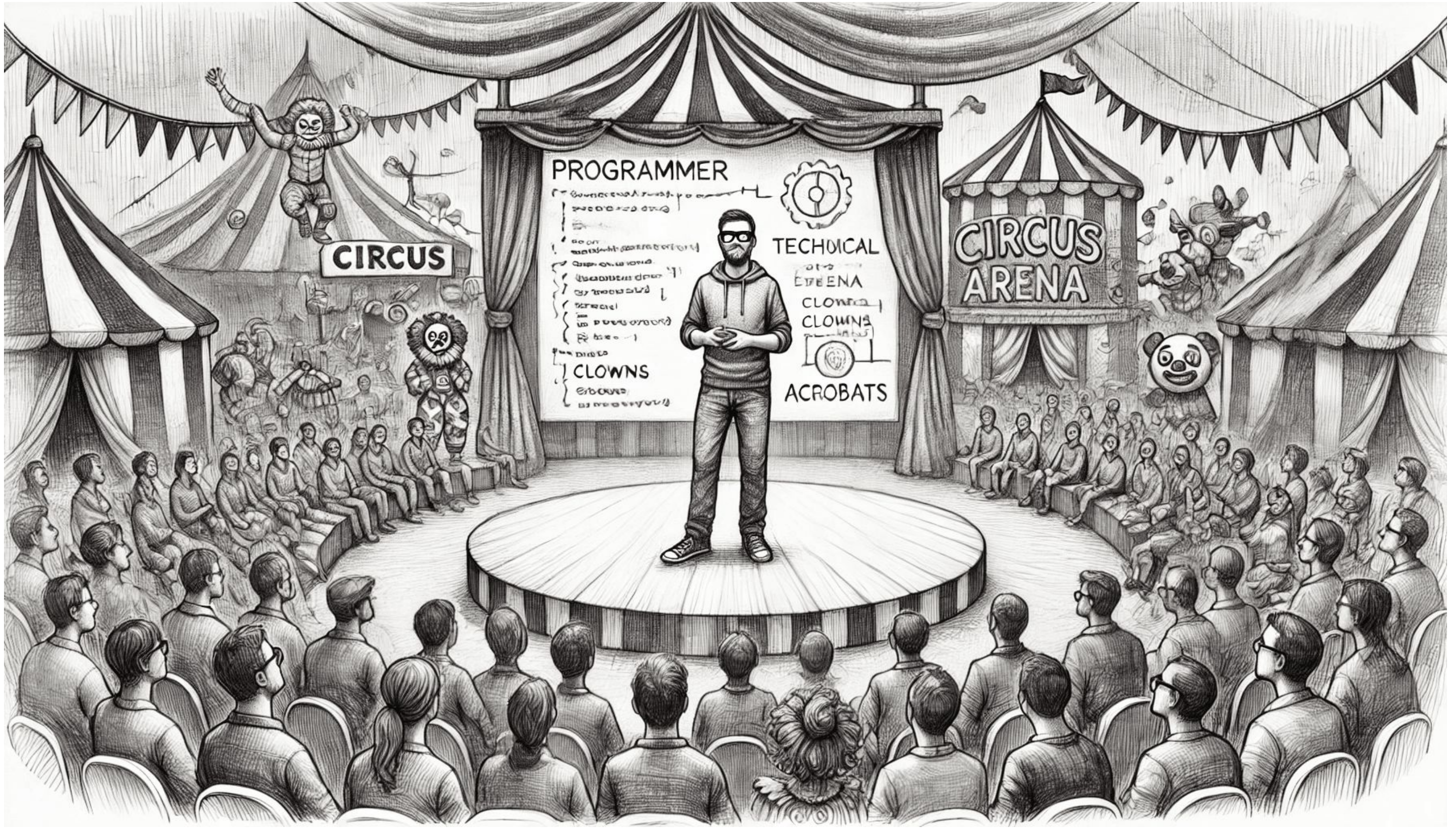
Preparation assistance (slide templates, suggested topic list, mentoring/feedback loop for the presentation)

part 1: philosophical outlook

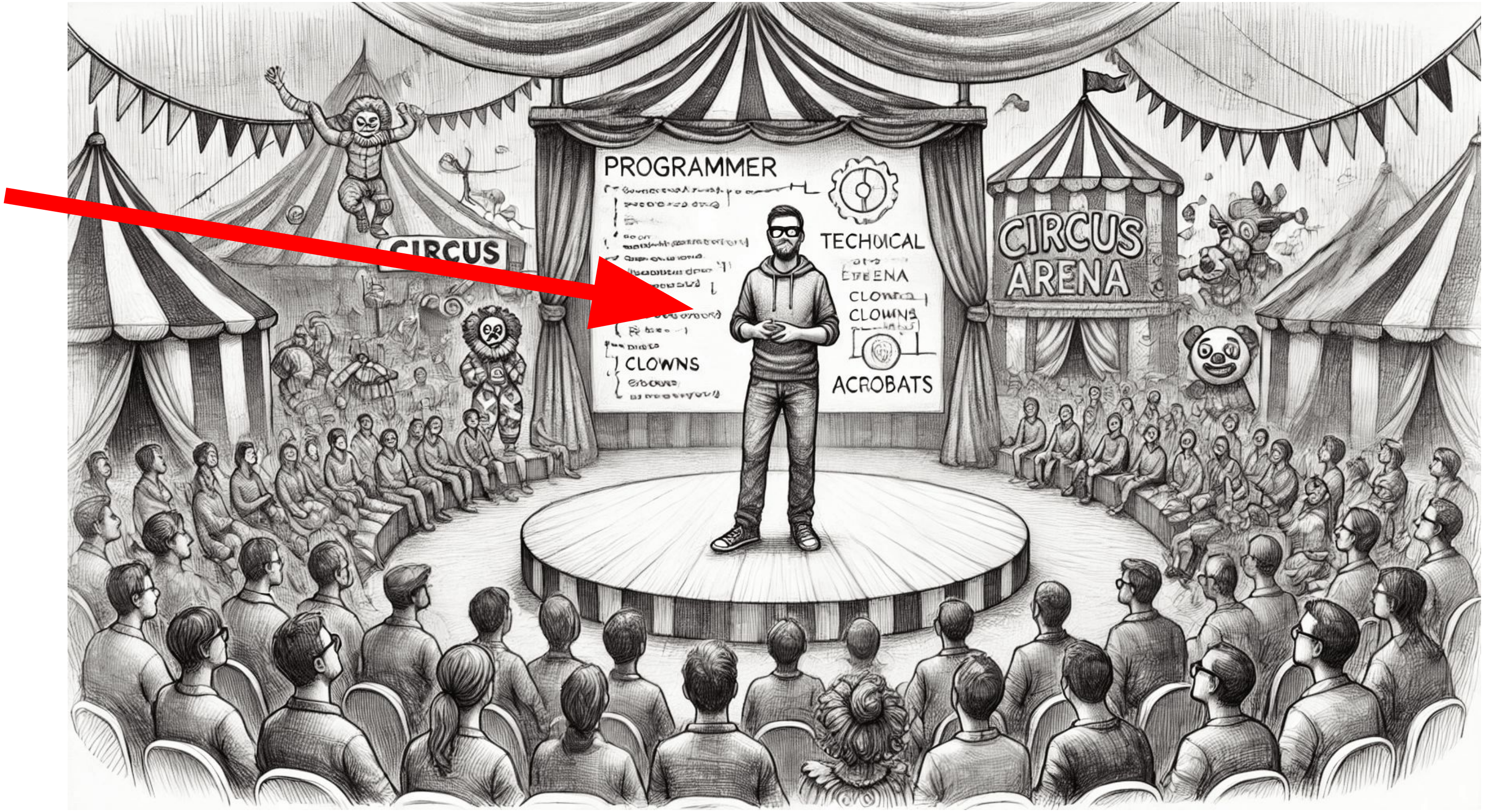


reality

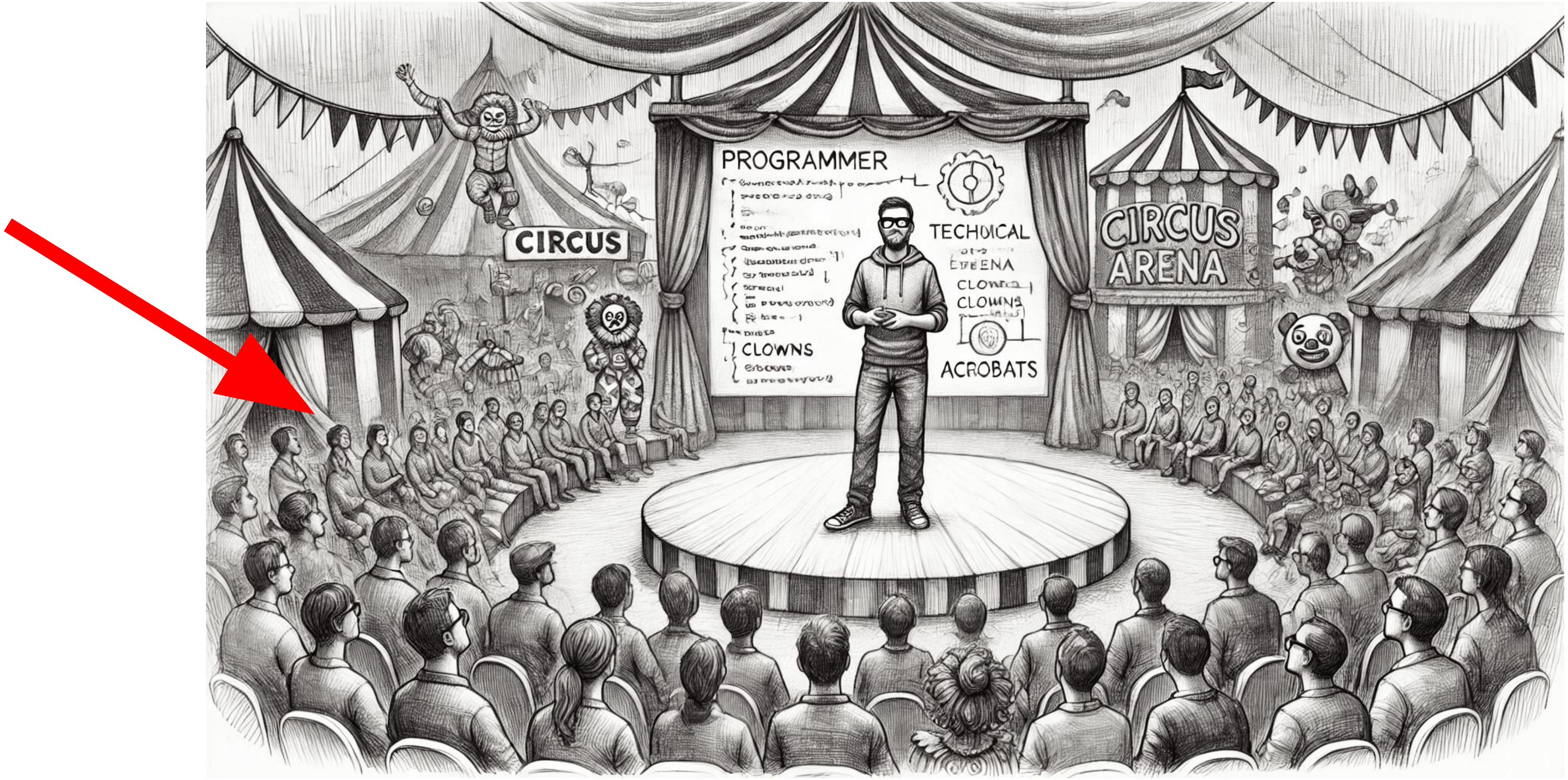
why learn presentation?



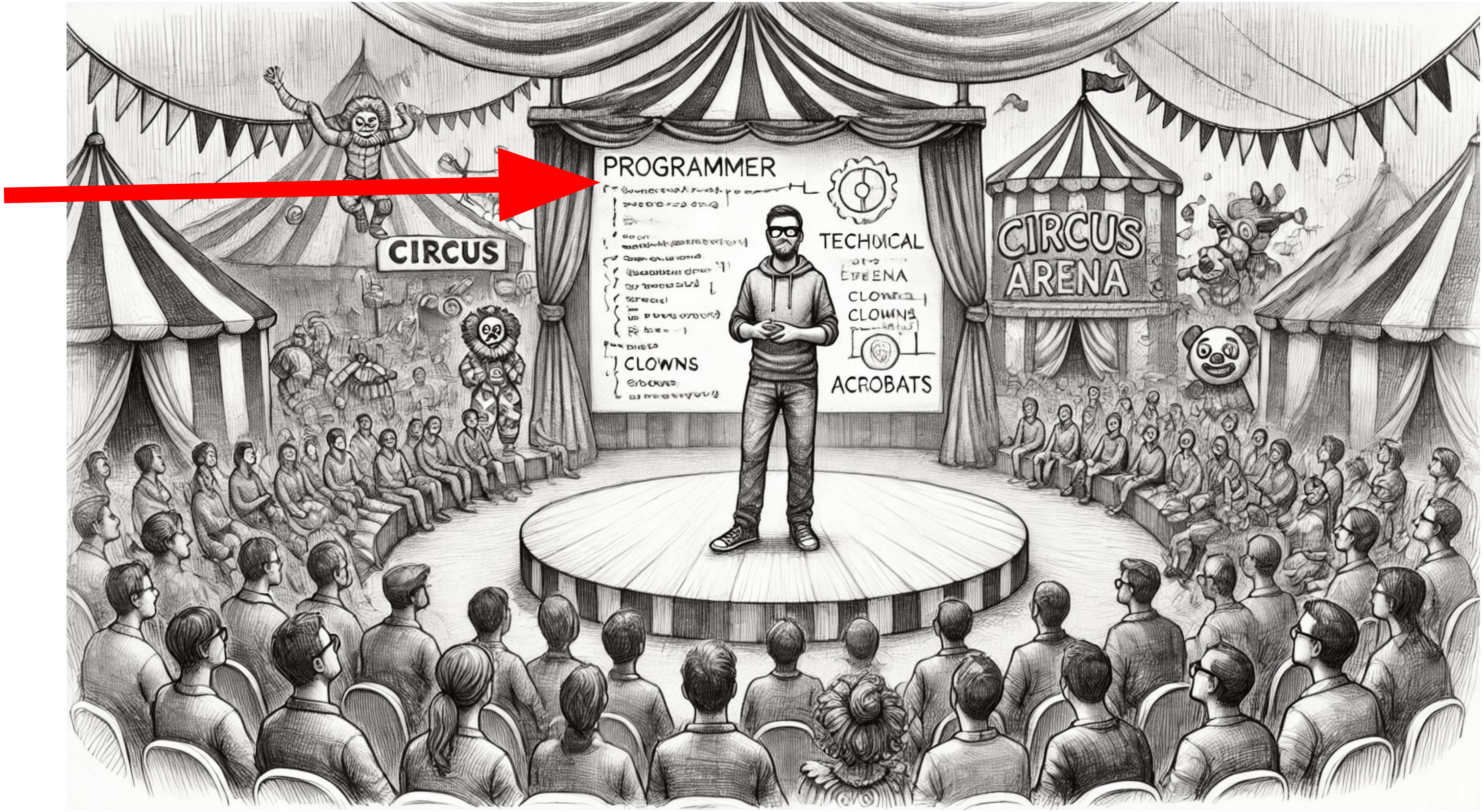
world is a circus and I am the clown



world is a circus and I am the clown



world is a circus and I am the clown



world is a circus and I am the clown

purpose

evoke emotions;

- hope

- engagement

- increased self-confidence

trigger actions;

- apply to give a talk

- start using programming pattern

- buy a service

- save the animals

fear is the mind killer

what could go wrong?

part 2: practical TODO list

step 1: what is the emotion you want for your audience?

step 2: generate a title

1. "Getting Started with Go: A Beginner's Guide"
2. "Concurrency in Go: Mastering Goroutines and Channels"
3. "Building Scalable Microservices with Go"
4. "Effective Error Handling in Go: Best Practices and Patterns"
5. "The Power of Go Modules: Dependency Management Simplified"
6. "Optimizing Performance in Go Applications"
7. "Unit Testing in Go: Tools and Techniques"
8. "Exploring the Go Standard Library: Hidden Gems"
9. "Real-time Web Applications with Go and WebSockets"
10. "Leveraging Go for Cloud-Native Development"
11. "Data Processing Pipelines in Go: An Introduction"
12. "Migrating Legacy Code to Go: Challenges and Strategies"

step 2: generate a title (but better)

1. "Deep Dive into Go's Garbage Collector: How It Works and How to Tune It"
2. "Mastering Goroutines: Patterns for Concurrent Programming in Go"
3. "Leveraging Go Interfaces for Flexible and Maintainable Code"
4. "The Intricacies of Go's Memory Management: Pointers, Allocation, and Performance"
5. "Building Robust APIs with Go: Best Practices and Advanced Techniques"
6. "Exploring Go's Context Package: Managing Deadlines, Cancellation, and Timeouts"
7. "Advanced Channel Patterns in Go: Pipeline, Fan-Out, and Fan-In"
8. "Understanding Go's Package Structure and Module System"
9. "Effective Use of Reflection in Go: When and How to Use It"
10. "Error Handling Strategies in Go: From Basic to Advanced"
11. "Optimizing Go Applications: Profiling, Benchmarking, and Performance Tuning"
12. "Implementing Secure Authentication and Authorization in Go Web Applications"

step 3: write a prospect for the story behind the title

- hardest part

- you probably need less than you generate, but it's important to generate as much as you can

- all ideas are good initially

- writing, rewriting, dropping ideas and coming up with new ones is a process

- it takes time

step 4: apply time constraints

1. length of talk is finite (chatgpt suggests 10-15 slides for a 15 minute presentation, but I've got 26)

2. your time to prepare a talk is finite (start late, then you're constrained by reality)

the less time you give yourself to prepare content, the better you have to be at beating around the bush

step 5: fill the content

step 5: do work that's good enough

use ai

use google docs

use go present if you feel like it

use public code

write your own samples

take code from work (ASK FOR PERMISSION USING EMAIL FIRST)

step 6: execute your plan

step 7: flex on your haters

you just gave a public speech

you're the authority figure now

if people criticize you just say

> I'd love to hear a talk from you with your opinion

and then they never do so you're safe

thank you for your attention

please submit your talk