



Multi-client runtime generation with ASP.NET

Georgi Karagov | 31/07/2021

Georgi Karagogov



Geodetic Engineer (UACEG & Ghent University)

Full Stack Software Engineer @ Scalefocus

Founder and co-founder of a few products

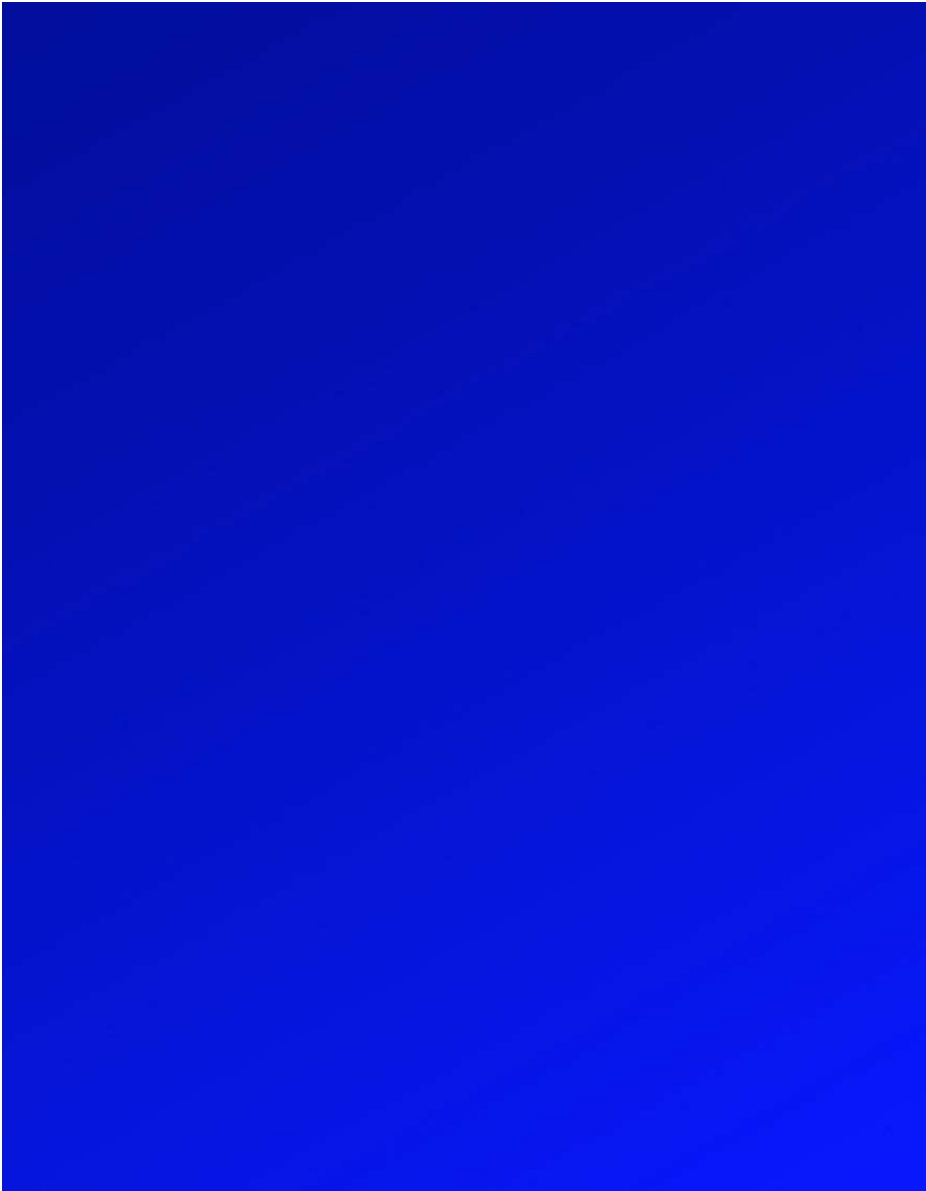
Creator and maintainer of Definux Open Source

UX/UI & Marketing enthusiast





Agenda

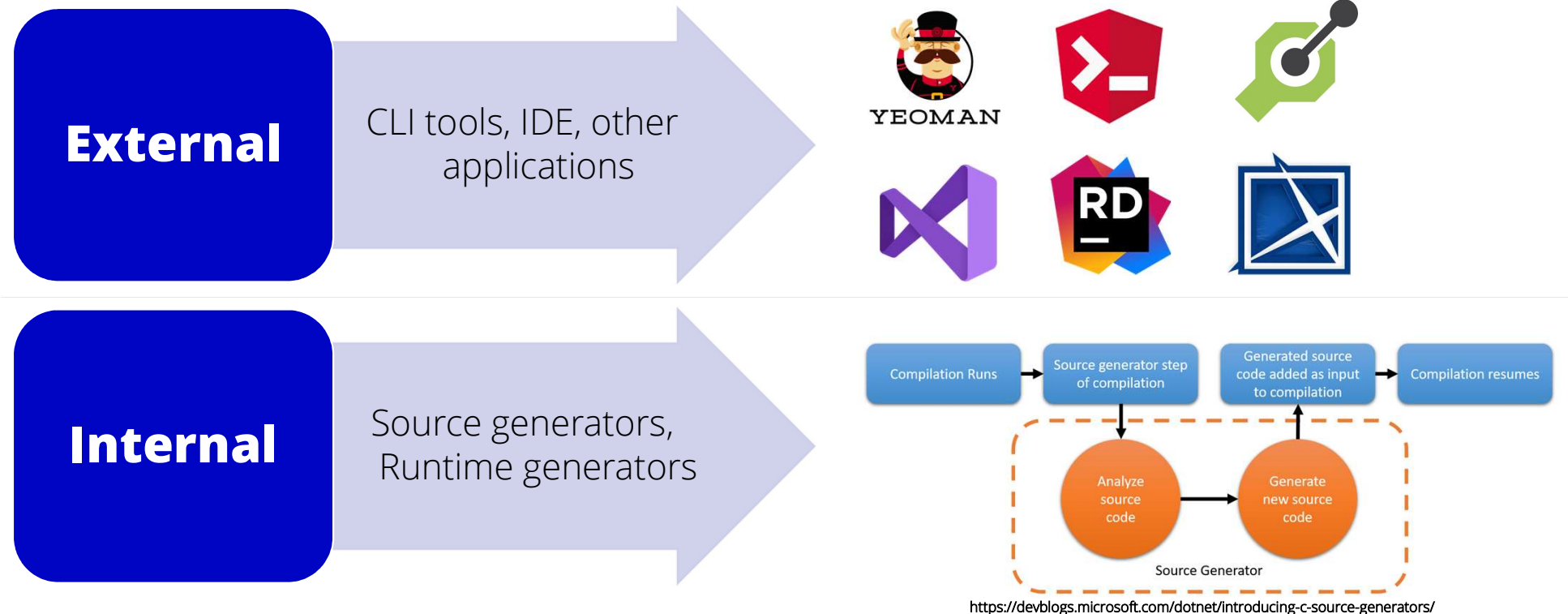
- 01 | **Source-Code generation**
 - 02 | **Source entities**
 - 03 | **Modules definition**
 - 04 | **Generation accessibility**
 - 05 | **Client consuming**
 - 06 | **Demo**
 - 07 | **Q&A**
- 

Source-Code Generation

Source-Code generation is the process of creating source code based on a description of a problem or template with contextually related data.



Types | Source-Code Generation



Source-Code Runtime Generation

Source
Entities

Modules and Modules
Factory

Accessibility Layer

Generated Source
Code



Source Entities

Assemblies	Databases	Internal files	External files	Other
classes, types, constants, enumerations, instances, etc.	schema, data, etc.	configuration, assets, docs, etc.	generators results, application results, etc.	helper applications, third-party providers, etc.



Data Extraction | Source Entities

Naming conventions

DogsController,
UserService

Interfaces

IEditViewModel,
IService

Inheritance

CrudController,
CreateEditDto

Decoration

[Required], [HttpGet]

Namespacing

.Controllers,
.Services

Custom references

.Add(typeof(UserDto));



Modules Definition

GeneratationProvider
- modules
+ Generate () : void

Module
- id - type - options
+ GetId () : int + Generate () : void # SetupFolders () : void # SetupFiles () : void

ModuleFolder
- id - name - path
+ GetPath () : string

ModuleFile
- id - name - path - template
+ GetPath () : string + Render () : string
Template



Modules Definition

```
public override void SetupFolders()
{
    AddFolder(new ModuleFolder
    {
        Name = "folder1",
        RelativePath = Options.ClientAppPath,
    });

    AddFolder(new ModuleFolder
    {
        Name = "folder2",
        RelativePath = Options.ClientAppPath,
    });
}
```



Modules Definition

```
public override void SetupFiles()
{
    string relativePath = Path.Combine(Options.ClientAppPath, "folder", "subfolder");

    AddFile(new ModuleFile
    {
        Name = "file1.js",
        RelativePath = relativePath,
        TemplateType = typeof(File1Template),
        RenderFunction = RenderFile1
    });

    AddFile(new ModuleFile
    {
        Name = "file2.js",
        RelativePath = relativePath,
        TemplateType = typeof(File2Template),
        RenderFunction = RenderFile2
    });
}
```



Generation Accessibility

Referenced project

single project or
abstractions package +
few implementation
projects

Development environment only

loading only during
development

Isolated from the core application/s

BLL doesn't depend
on generation
assembly

Simple triggering

HTTP requests or by
using project
initialization method

Client side if necessary

simple page with action
buttons and status
information



Client Consuming

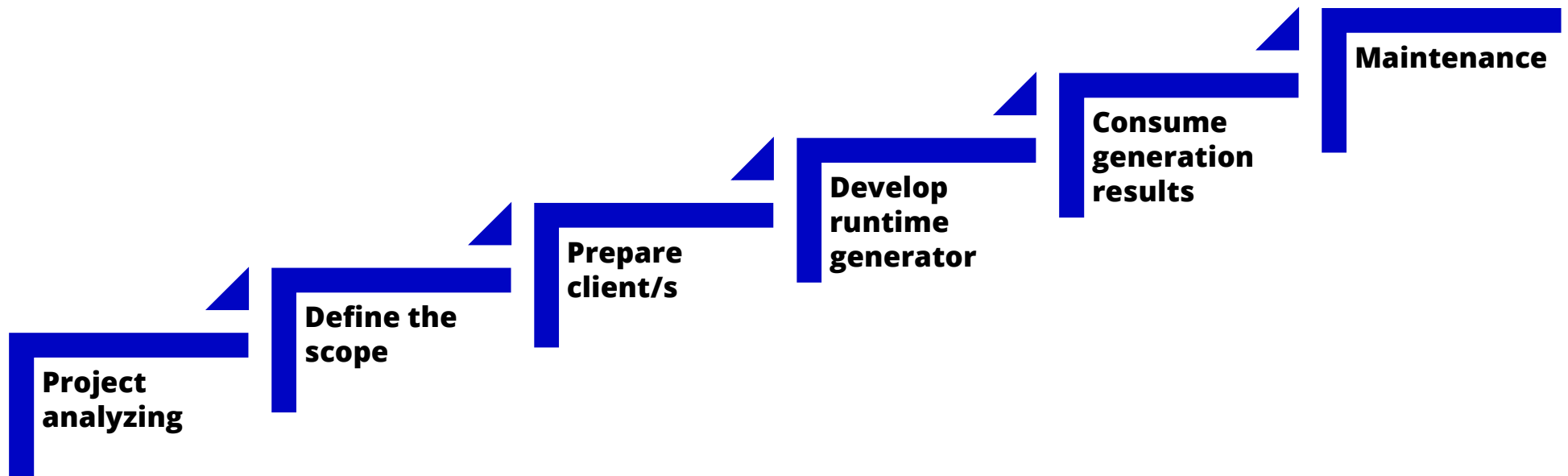




Demo

<https://github.com/gsk567/Scalefocus-BurgasConf2021>

Summary





Q&A



Thank you

📞 +359 2424 6484 | ✉ sales@scalefocus.com | 🌐 www.scalefocus.com