

Streamlining application development for Genode with Goa



Johannes Schlatow

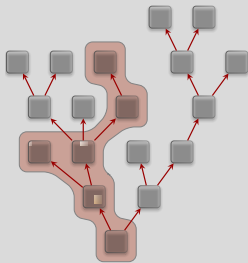
`<johannes.schlatow@genode-labs.com>`



What is Genode?

Genode OS Framework

- component-based OS framework
- supports different (micro-)kernels and CPU architectures
- quarterly releases since 2008

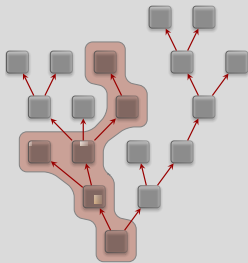




What is Genode?

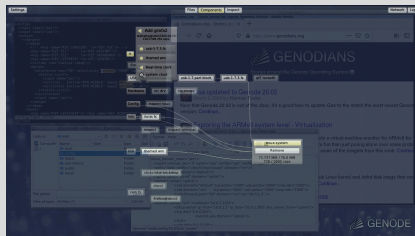
Genode OS Framework

- component-based OS framework
- supports different (micro-)kernels and CPU architectures
- quarterly releases since 2008



Sculpt OS

- OS showcase based on Genode
- used as daily driver by developers
- targets PC and PinePhone
- releases twice a year





Why Goa?

Genode's build system

- useful and efficient for framework development
- cumbersome for app development (framework users)
- steep learning curve → off-putting for newcomers and hobbyists



Why Goa?

Genode's build system

- useful and efficient for framework development
- cumbersome for app development (framework users)
- steep learning curve → off-putting for newcomers and hobbyists

Goa - *goal, but reached a little bit sooner*

- started in 2019 by Norman Feske as a side project
- moved under the umbrella of Genode Labs in 2023
- has seen quite a few feature additions since then
- streamlines development of individual applications



How are applications made available to other users?

Genode Package Management

- each user manages its own **depot**
- a depot contains different types of **archives**:
 - api** header files
 - src** source code
 - bin** architecture-specific binary for *src* archive
 - raw** raw data archive, architecture-independent
 - pkg** runtime information, lists required raw/src/pkg archives
 - index** curated list of a user's pkg archives, shown in Sculpt OS
- archives are placed in corresponding **subdirectories** within depot



How are applications made available to other users?

Genode Package Management

- each user manages its own **depot**
- a depot contains different types of **archives**:
 - `api` header files
 - `src` source code
 - `bin` architecture-specific binary for `src` archive
 - `raw` raw data archive, architecture-independent
 - `pkg` runtime information, lists required `raw/src/pkg` archives
 - `index` curated list of a user's `pkg` archives, shown in Sculpt OS
- archives are placed in corresponding **subdirectories** within depot

→ a Goa project resembles this structure



Goa workflow





Goa workflow



```
$ goa [import,build,run,export,publish,...]
```



Importing 3rd-party source code

```
$ goa import
```

- requires *import* file in project directory
- populates *src* and/or *raw* directory

```
«project-name»/
```

```
└─import
```



Importing 3rd-party source code

```
$ goa import
```

- requires *import* file in project directory
- populates *src* and/or *raw* directory

```
«project-name»/
```

```
└─import
```

```
LICENSE    := GPLv3  
VERSION    := 3.3.7  
DOWNLOADS := calc.archive
```

```
BASE_URL   := https://gitlab.com/ubports/development/apps/lomiri-calculator-app/-/archive/  
URL(calc)  := $(BASE_URL)/v$(VERSION)/lomiri-calculator-app-v$(VERSION).tar.gz  
SHA(calc)  := 821f045...  
DIR(calc)  := src
```

See also: `goa help import`



Importing 3rd-party source code

```
$ goa import
```

- requires *import* file in project directory
- populates *src* and/or *raw* directory

```
«project-name»/
```

```
└─import
```

```
LICENSE := GPLv3  
VERSION := 3.3.7  
DOWNLOADS := calc.archive
```

supported types: archive, git, svn, file

```
BASE_URL := https://gitlab.com/ubports/development/apps/lomiri-calculator-app/-/archive/  
URL(calc) := $(BASE_URL)/v$(VERSION)/lomiri-calculator-app-v$(VERSION).tar.gz  
SHA(calc) := 821f045...  
DIR(calc) := src
```

See also: `goa help import`



Importing 3rd-party source code

`$ goa import`

- requires *import* file in project directory
- populates *src* and/or *raw* directory

```
«project-name»/  
└─ import
```

```
LICENSE    := GPLv3  
VERSION    := 3.3.7  
DOWNLOADS := calc.archive
```

supported types: archive, git, svn, file

```
BASE_URL   := https://gitlab.com/ubports/development/apps/lomiri-calculator-app/-/archive/  
URL(calc)  := $(BASE_URL)/v$(VERSION)/lomiri-calculator-app-v$(VERSION).tar.gz  
SHA(calc)  := 821f045...  
DIR(calc)  := src
```

"src" or "raw"

See also: `goa help import`



Building software using commodity build systems

`$ goa build`

- compiles code in *src/* directory
- requires *used_apis* and *artifacts* file

«project-name»/

src/

used_apis

artifacts

See also: `goa help [artifacts|build-systems]`



Building software using commodity build systems

`$ goa build`

- compiles code in *src/* directory
- requires *used_apis* and *artifacts* file

supported build systems

- plain GNU Make
- CMake
- autoconf
- qmake
- cargo (Rust)
- *...add your favorite build system*

«project-name»/

src/

used_apis

artifacts

configure_args

make_args

cmake_args

See also: `goa help [artifacts|build-systems]`



Test-Running a scenario

```
$ goa run
```

required files

- *pkg/«name»/archives*
- *pkg/«name»/runtime*



See also: `goa help [runtime|targets]`



Test-Running a scenario

```
$ goa run
```

required files

- *pkg/«name»/archives*
- *pkg/«name»/runtime*

supported targets (via --target)

- *linux* (host system, default)
- *sculpt* (file sync via HTTP PUT)
- *...add your own target*

See also: `goa help [runtime|targets]`





Demo

Demo



Exporting and Publishing

\$ goa publish (*implies export*)

- export: assembles depot archives
- publish: creates signed tar archives

See also: goa help add-depot-user



Exporting and Publishing

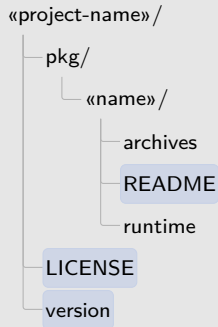
\$ `goa publish` (*implies export*)

- `export`: assembles depot archives
- `publish`: creates signed tar archives

required files

- `version`
- `LICENSE`
- `pkg/«name»/README`

See also: `goa help add-depot-user`






Demo

Demo














Getting Started


User stories / Tutorials





Stories around the Genode Operating System


Authors


-  Johannes Schlatow
Genode Labs
-  Josef Söntgen
Genode Labs
-  Pirmin Duss
-  Norman Feske
Genode Labs
-  Benjamin Lamowski
Genode Labs
-  Martin Stein
Genode Labs
-  Alexander Böttcher
Genode Labs
-  Stefan Kalkowski
Genode Labs
-  Sebastian Sumpf
Genode Labs
-  Michael Grunditz
-  Christian Helmuth

 **Goa - Test running applications on Sculpt**
January 29 2024 by Johannes Schlatow
Since its first release, Goa allows test-running applications on the host. We recently added an alternative run target to Goa which enables executing a Goa application on a Sculpt system. [Continue...](#)

 **Goa - Porting a calculator app from Ubuntu UI Toolkit**
January 11 2024 by Johannes Schlatow
I ported the calculator app from Ubuntu/Lomiri UI Toolkit. In this article, I walk you through the porting procedure step-by-step to provide a blueprint for porting other apps. It also serves as an additional and more advanced Goa tutorial. [Continue...](#)

 **Manually installing wsman(1)**
January 5 2024 by Josef Söntgen
For convenience and to some degree also necessity most of our x86-based test machines feature AMT to power-cycle the system and also gain serial console access via SOL. On recent systems the controlling part is done via WS-MAN, for which a package is not always readily available on Linux distributions. The following post gives short instructions on how to build it manually and illustrates one or the other pitfall. [Continue...](#)

 **A SIP client for Genode on the PinePhone**
November 16 2023 by Josef Söntgen
We ported the "Lirphone Console Client" and the "SIP Client for Ubuntu Touch" to Genode to expand the available features on the PinePhone when it comes to mobile communication. This article illustrates the integration and describes how you can experience it yourself. [Continue...](#)

 **Create a specialized run target for Goa**
November 14 2023 by Pirmin Duss
If you need special services for your Goa project, you can easily create a customized run target for testing your project. I will describe how you can achieve this in this article with two simple examples. [Continue...](#)

External Links

- Genode OS project
- Genode at GitHub
- Join genodians.org
- Discuss at reddit
- Mailing list
- IRC

<https://genodians.org>



Getting Started

User stories / Tutorials

GENODIANS
Stories around the Genode Operating System

Authors

- Johannes Schlatow
Genode Labs
- Josef Söntgen
Genode Labs
- Pirmin Duss

External Links

- Genode OS project
- Genode at GitHub
- Join genodians.org
- Discuss at reddit
- Mailing list
- IRC

Thank you!

```
$ git clone https://github.com/genodelabs/goa
```

Sebastian Sumpf
Genode Labs

Michael Grunditz

Christian Helmuth

<https://genodians.org>



Goa source code

<https://github.com/genodelabs/goa>

Goa project repositories

<https://github.com/nfeske/goa-playground>

<https://github.com/{jschlatow,cnuke,ssumpf}/goa-projects>

Genodians.org community blog

<https://genodians.org>

Genode OS Framework

<https://genode.org>

Sculpt OS download and manual

<https://genode.org/download/sculpt>