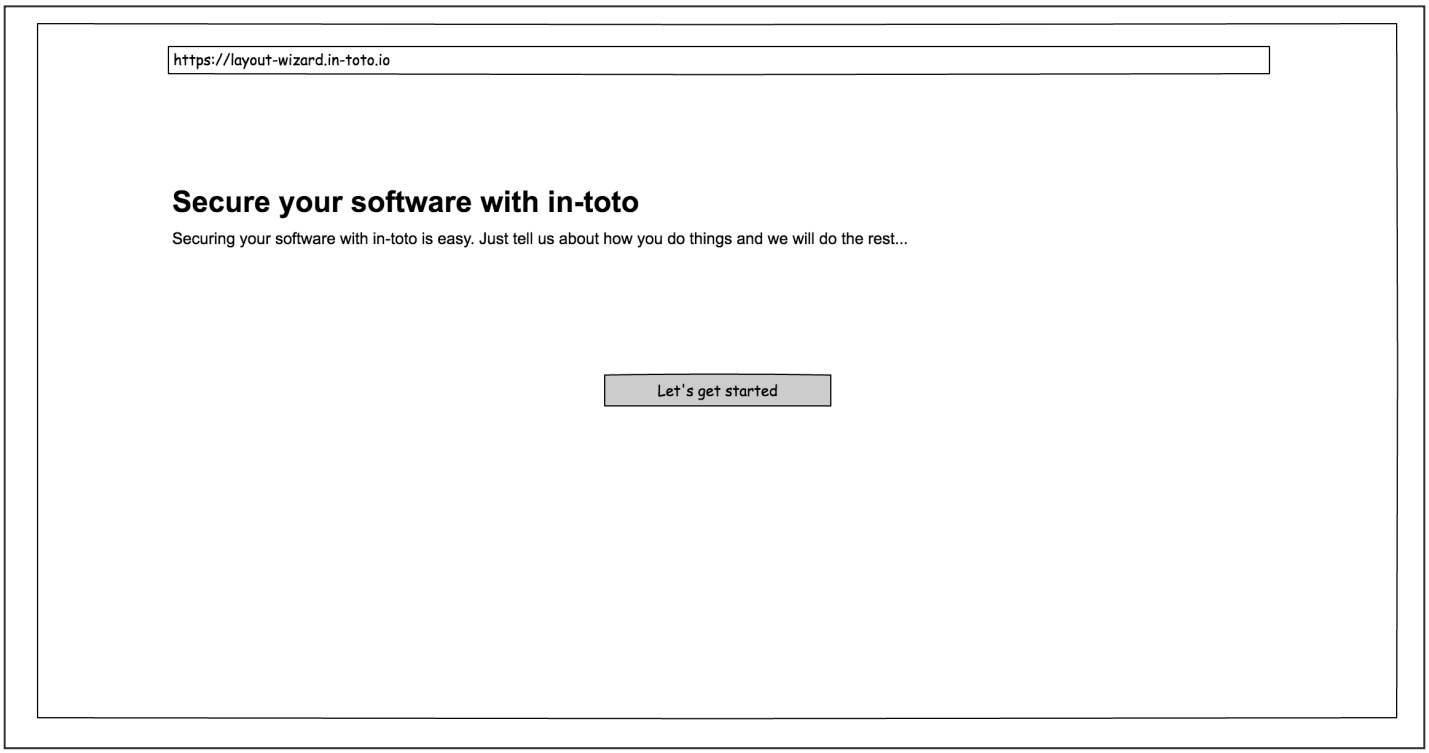
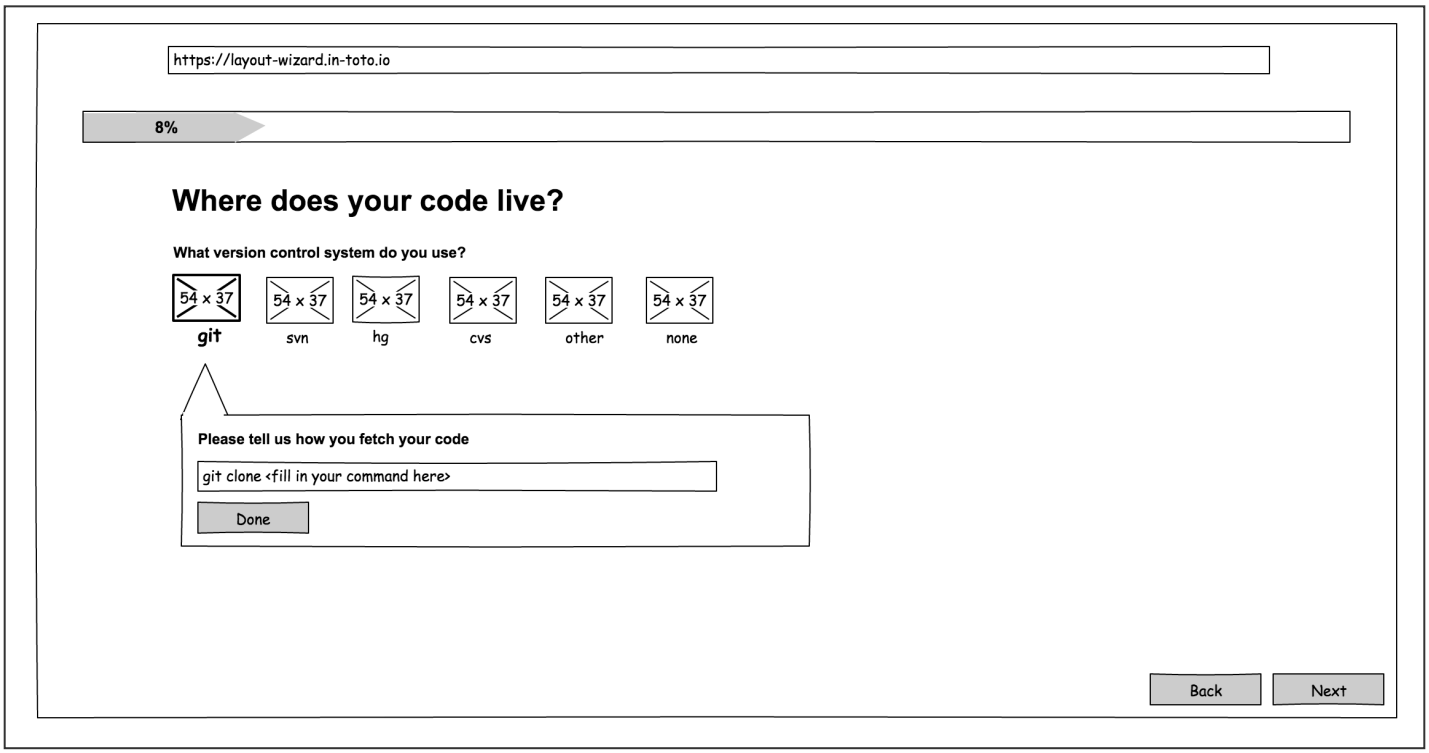
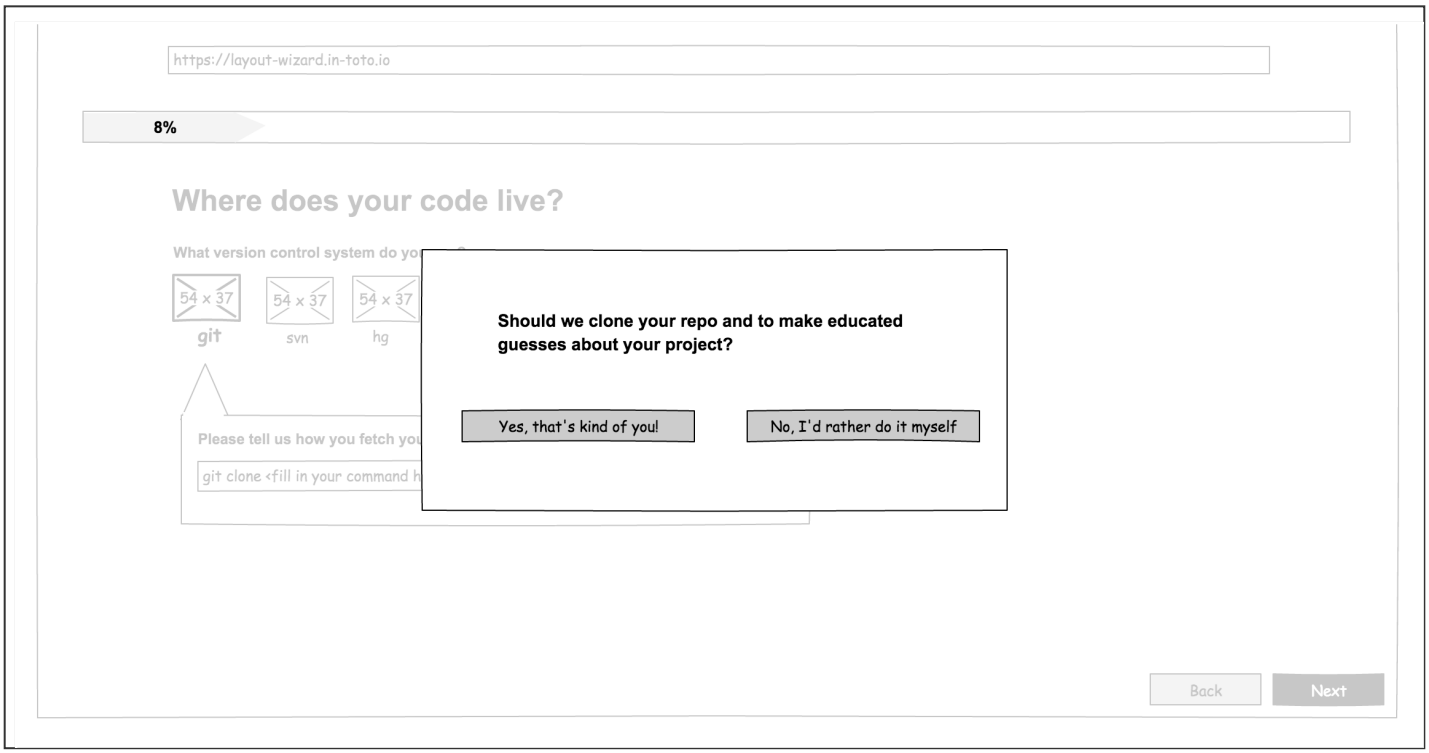
start


https://layout-wizard.in-toto.io
Secure your software with in-toto
Securing your software with in-toto is easy. Just tell us about how you do things and we will do the rest...
Let's get started

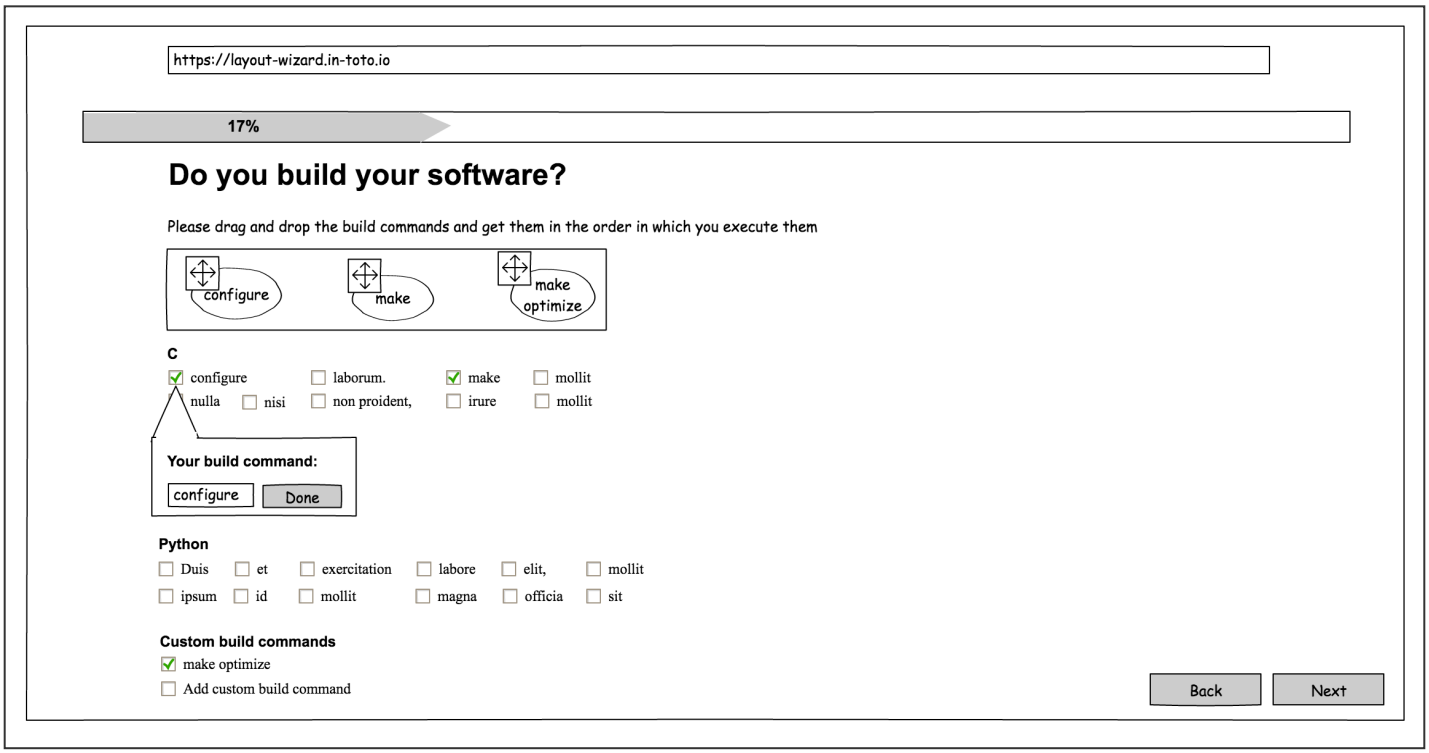
vcs


https://layout-wizard.in-toto.io
8%
Where does your code live?
What version control system do you use?
54 x 37
git
54 x 37
svn
54 x 37
hg
54 x 37
cvs
54 x 37
other
54 x 37
none
Please tell us how you fetch your code
git clone <fill in your command here>
Done
Back
Next

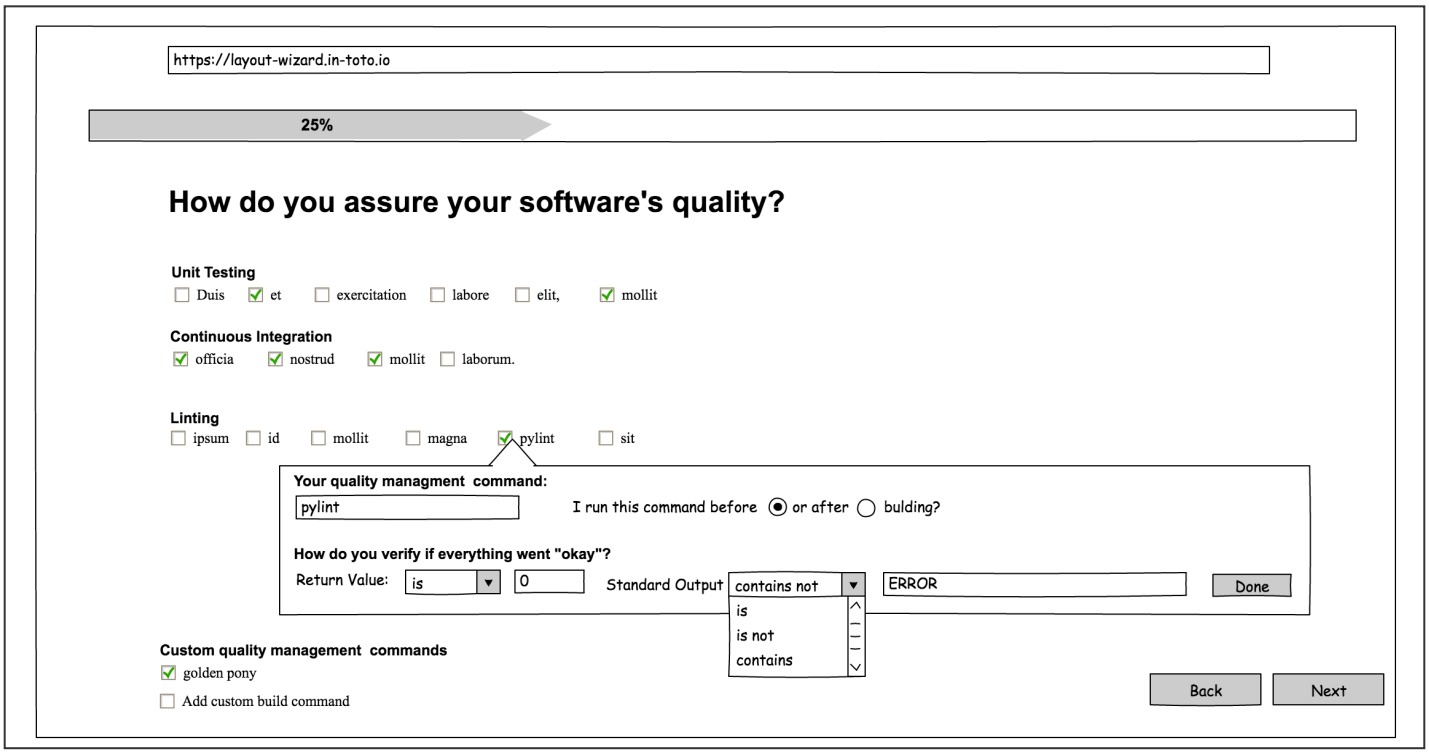
vcs (1)


https://layout-wizard.in-toto.io
8%
Where does your code live?
What version control system do yo
54 x 37
54 x 37
54 x 37
git
svn
hg
Please tell us how you fetch you
git clone <fill in your command h
Should we clone your repo and to make educated guesses about your project?
Yes, that's kind of you!
No, I'd rather do it myself
Back
Next

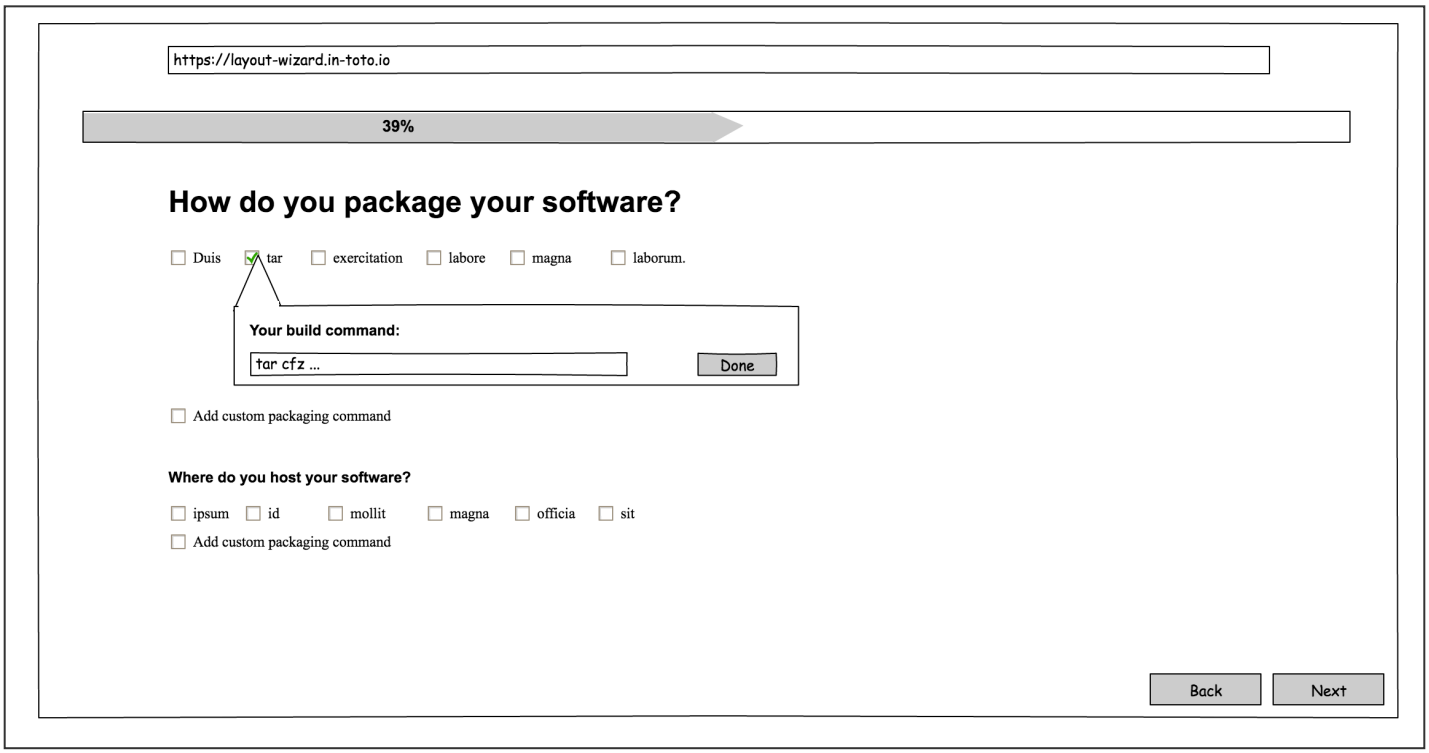
building


https://layout-wizard.in-toto.io
17%
Do you build your software?
Please drag and drop the build commands and get them in the order in which you execute them
configure
make
make optimize
C
configure
laborum.
make
mollit
nulla
nisi
non proident,
irure
mollit
Your build command:
configure
Done
Python
Duis
et
exercitation
labore
elit,
mollit
ipsum
id
mollit
magna
officia
sit
Custom build commands
make optimize
Add custom build command
Back
Next

qm


https://layout-wizard.in-toto.io
25%
How do you assure your software's quality?
Unit Testing
Duis
et
exercitation
labore
elit,
mollit
Continuous Integration
officia
nostrud
mollit
laborum.
Linting
ipsum
id
mollit
magna
pylint
sit
Your quality managment command:
pylint
I run this command before
or after
bulding?
How do you verify if everything went "okay"?
Return Value:
is
0
Standard Output
contains not
ERROR
Done
is
is not
contains
Custom quality management commands
golden pony
Add custom build command
Back
Next

release

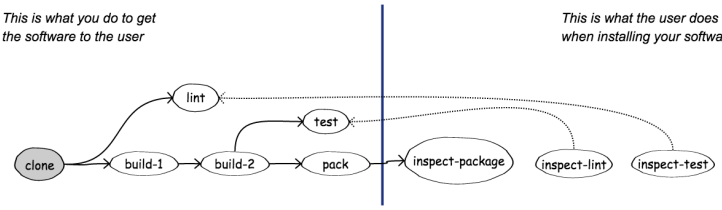

https://layout-wizard.in-toto.io
39%
How do you package your software?
Duis
tar
exercitation
labore
magna
laborum.
Your build command:
tar cfz ...
Done
Add custom packaging command
Where do you host your software?
ipsum
id
mollit
magna
officia
sit
Add custom packaging command
Back
Next

tweak steps

https://layout-wizard.in-toto.io

43%

Is this what your software supply chain looks like?



Are we assuming correctly that you first **clone** your software project from mygitproject.git with git, then run **lint** and build it using **configure** and **make**, and then **test** the built software with the command runtests before for you **package** it as tar archive?

And is it true that if we **inspected the package** it should contain exactly what you put in there in your package step? And further is it right that the the **lint** command was successfull if it **returned 0** whereas the **test** command was successful if it **didn't have "ERROR"** in its output?

I should probably clarify some things

The steps of your software supply chain

Name: clone	Command:	git clone myproject.git	Remove
Name: lint	Command:	lint myproject	Remove
Name: build-1	Command:	configure myproject	Remove
Name: build-2	Command:	make myproject	Remove
Name: test	Command:	runtests myproject	Remove
Name: pack	Command:	tar czf myproject.tar.gz myproject	Remove

Add Step

The inspections of your software supply chain

Name: inspect-package	Command:	tar xzf myproject.tar.gz	Remove
Name: inspect-lint	Command:	in-toto-inspect --step lint --retval --equals 0	Remove
Name: inspect-test	Command:	in-toto-inspect --step test --stdout --contains-not "ERROR"	Remove

Add Inspection

Done

Back Next

keys

https://layout-wizard.in-toto.io

54%

Who is allowed to do what in your project?

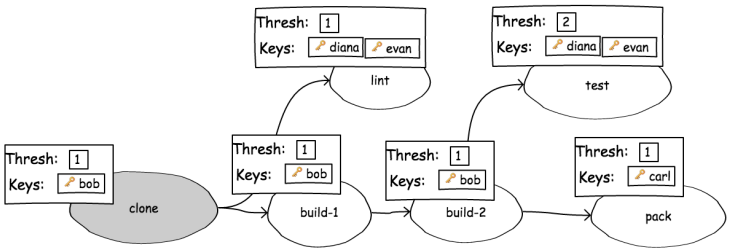
Upload Public Keys

Upload your "functionary" (the people who do the things) public keys. If you don't have any keys, you can easily generate them using our in-toto key generation tool.

bob.pub | carl.pub | diana.pub | evan.pub — Public Key Dropzone

Authorize your functionaries to perform steps

Drag and drop the uploaded keys from above onto the supply chain steps to authorize one or many functionaries to perform a step.
The threshold indicates how many functionaries are required that step.



Back | Next

dry run

https://layout-wizard.in-toto.io

78%

Security Restriction Assessment a.k.a "linking your supply chain"

Now that we have figured out what your software supply chain looks like we need just a little bit more information about the files you are working on in each step.

Just paste the following commands in a terminal and upload the resulting link file archive. We will use the information to generate a set of rules that chains the invidiual links of your supply chain together.

Copy/Paste Commands

```
$ in-toto-run --dry  --name clone -- git clone myproject.git
$ in-toto-run --dry  --name lint -- lint myproject
$ in-toto-run --dry --name build-1 -- configure myproject
$ in-toto-run --dry --name build-2 -- make myproject
$ in-toto-run --dry --name test -- runtests myproject
$ in-toto-run --dry --name package -- tar -czf myproject.tar.gz myproject

$ tar czf all-links.tar.gz clone.link lint.link build-1.link build-2.link test.link package.link
```

Upload "all-links.tar.gz"

Back Next

generate layout

https://layout-wizard.in-toto.io

99%

Congrats, you are now the proud owner of a custom in-toto software supply chain layout!

But not so fast you still have to do a couple of things before your clients can verify your software with in-toto!

1. Download the layout

You can study and edit it using any text editor - it is just a JSON.

Download Layout

2. Create a project owner key pair (if you don't already have one)

```
$ in-toto-keygen project-owner
```

3. Sign your layout with your project owner key

```
$ in-toto-sign --key project-owner root.layout
```

4. Instruct your functionaries to use in-toto commands

Bob

```
$ in-toto-run --key bob --materials . --products .  --step-name clone -- git clone myproject.git
$ in-toto-run --key bob --materials . --products .  --step-name build-1 -- configure myproject
$ in-toto-run --key bob --materials . --products .  --step-name build-2 -- make myproject
```

Diana

```
$ in-toto-run --key diana  --materials . --products . --record-byproducts --step-name lint -- lint myproject
$ in-toto-run --key diana --materials . --products . --record-byproducts --step-name test -- runtests myproject
```

Evan

```
$ in-toto-run --key evan --materials . --products . --record-byproducts --step-name test -- runtests myproject
```

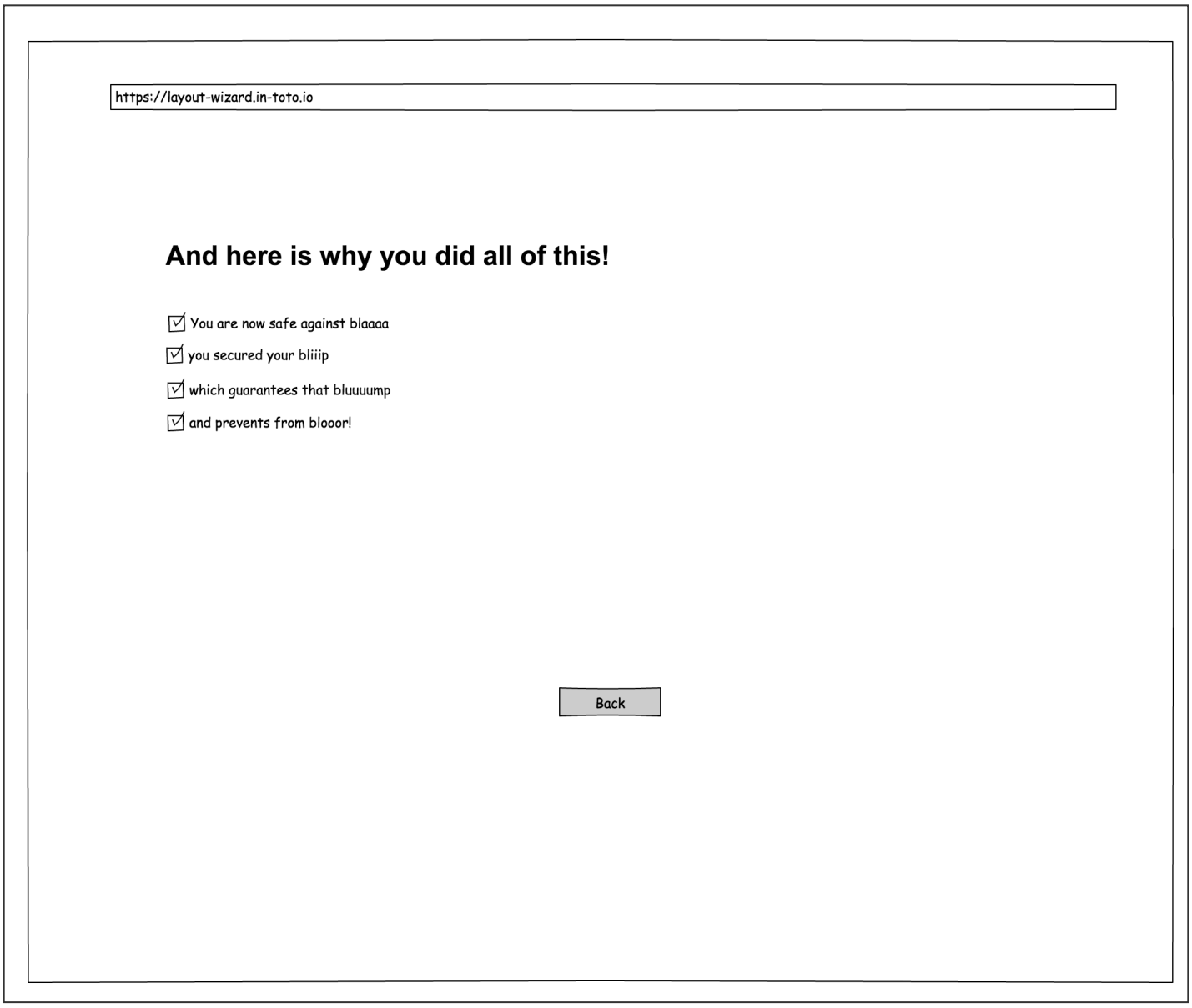
Carl

```
$ in-toto-run --key carl --materials . --products .  --step-name package -- tar -czf myproject.tar.gz myproject
```

5. Ship out in-toto metadata along with your software

Back

Curious to see how your supply chain is protected now?


https://layout-wizard.in-toto.io
And here is why you did all of this!
☑ You are now safe against blaaaa
☑ you secured your bliiip
☑ which guarantees that bluuuump
☑ and prevents from blooor!
Back