Nix and legacy enterprise software development: an unlikely match made in heaven

Maximilian (Ma27) Bosch

NixCon 2022

Intro

- We never used Nix in this project (that's a different story).
- The colleagues in this project were rather skeptical about Nix.
- This is the story of how I sneaked it in at one occasion.

Intro

- We never used Nix in this project (that's a different story).
- The colleagues in this project were rather skeptical about Nix.
- This is the story of how I sneaked it in at one occasion.

Intro

- We never used Nix in this project (that's a different story).
- The colleagues in this project were rather skeptical about Nix.
- This is the story of how I sneaked it in at one occasion.

Backstory

- Large software for creating maintenance manuals.
- To be migrated to a new system.
- There's this image editor
  - ...written in Flash
  - ...couldn't recompile and didn't understand the code.
  - ...had to reverse-engineer the behavior for the migration.

Backstory

- Large software for creating maintenance manuals.
- To be migrated to a new system.
- There's this image editor
  - ...written in Flash
  - ...couldn't recompile and didn't understand the code.
  - ...had to reverse-engineer the behavior for the migration.

Backstory

- Large software for creating maintenance manuals.
- To be migrated to a new system.
- There's this image editor
  - ...written in Flash
  - ...couldn't recompile and didn't understand the code.
  - ...had to reverse-engineer the behavior for the migration.

Backstory

- Large software for creating maintenance manuals.
- To be migrated to a new system.
- There's this image editor
  - ...written in Flash
  - ...couldn't recompile and didn't understand the code.
  - ...had to reverse-engineer the behavior for the migration.

Backstory

- Large software for creating maintenance manuals.
- To be migrated to a new system.
- There's this image editor
  - ...written in Flash
  - ...couldn't recompile and didn't understand the code.
  - ...had to reverse-engineer the behavior for the migration.

Backstory

- Large software for creating maintenance manuals.
- To be migrated to a new system.
- There's this image editor
  - ...written in Flash
  - ...couldn't recompile and didn't understand the code.
  - ...had to reverse-engineer the behavior for the migration.

Backstory

Problem

It's 2021 and **Flash is dead.**

Backstory

Problem

It's 2021 and **Flash is dead.**

Question

How to revive Flash in 2021?

Here comes the magic

- Build a VM with **nixos-build-vms(8)**
- Install a **firefox** from NixOS 16.09 with flash support into it.

Here comes the magic

- Build a VM with **nixos-build-vms(8)**
- Install a **firefox** from NixOS 16.09 with flash support into it.

Here comes the magic

```
{
  vm = { pkgs, ... }: {
    environment.systemPackages = [
      (import
        (builtins.fetchTarball nixos1609tarball)
        { config.firefox.enableAdobeFlash = true; }
      ).firefox
    ];
    virtualisation.memorySize = 8192;
    # config for an xserver
  };
}
```

Here comes the magic

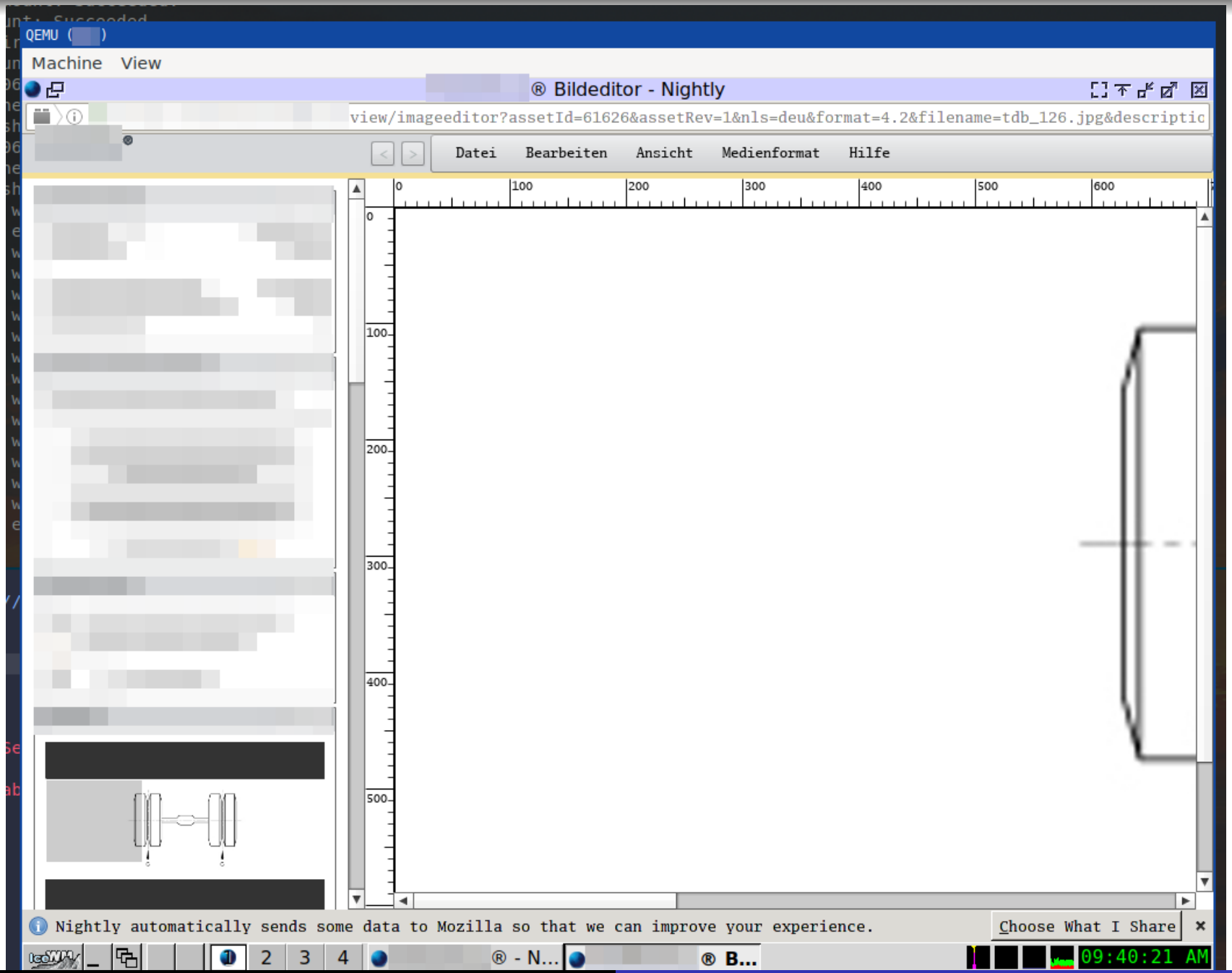
```
building '...'...

trying https://fpdownload.macromedia.com/pub/flashplayer
curl: (22) The requested URL returned error: 403
error: cannot download fp_11.2.202.635_archive.zip
from any mirror
```

Here comes the magic

```
nix-prefetch-url \
  'https://web.archive.org/web/20190628080005if_'\
  '/https://fpdownload.macromedia.com/pub/flashplayer/'\
  'installers/archive/fp_11.2.202.635_archive.zip'
```

Here comes the magic



Recap

- I hope you'll **never** need this!
- It's trivial to revive old applications because:
  - We have a binary cache with the build artifacts
  - We can instantiate **any** old Nix expression
  - Let's keep it that way!
- $Colleague was impressed by what Nix can do.

Recap

- I hope you'll **never** need this!
- It's trivial to revive old applications because:
  - We have a binary cache with the build artifacts
  - We can instantiate **any** old Nix expression
  - Let's keep it that way!
- $Colleague was impressed by what Nix can do.

Recap

- I hope you'll **never** need this!
- It's trivial to revive old applications because:
  - We have a binary cache with the build artifacts
  - We can instantiate **any** old Nix expression
  - Let's keep it that way!
- $Colleague was impressed by what Nix can do.

Recap

- I hope you'll **never** need this!
- It's trivial to revive old applications because:
  - We have a binary cache with the build artifacts
  - We can instantiate **any** old Nix expression
  - Let's keep it that way!
- $Colleague was impressed by what Nix can do.

Recap

- I hope you'll **never** need this!
- It's trivial to revive old applications because:
  - We have a binary cache with the build artifacts
  - We can instantiate **any** old Nix expression
  - Let's keep it that way!
- $Colleague was impressed by what Nix can do.

Recap

- I hope you'll **never** need this!
- It's trivial to revive old applications because:
  - We have a binary cache with the build artifacts
  - We can instantiate **any** old Nix expression
  - Let's keep it that way!
- $Colleague was impressed by what Nix can do.