DIRECTIONS 4 PARTNERS

Create your own Azure DevOps web extension

Tobias Fenster, COSMO CONSULT

www.directions4partners.com

Tobias Fenster

CTO at COSMO CONSULT Group

Dual Microsoft MVP for Business Applications and Azure

Microsoft Regional Director

@tobiasfenster

tobiasfenster.io

tobiasfenster

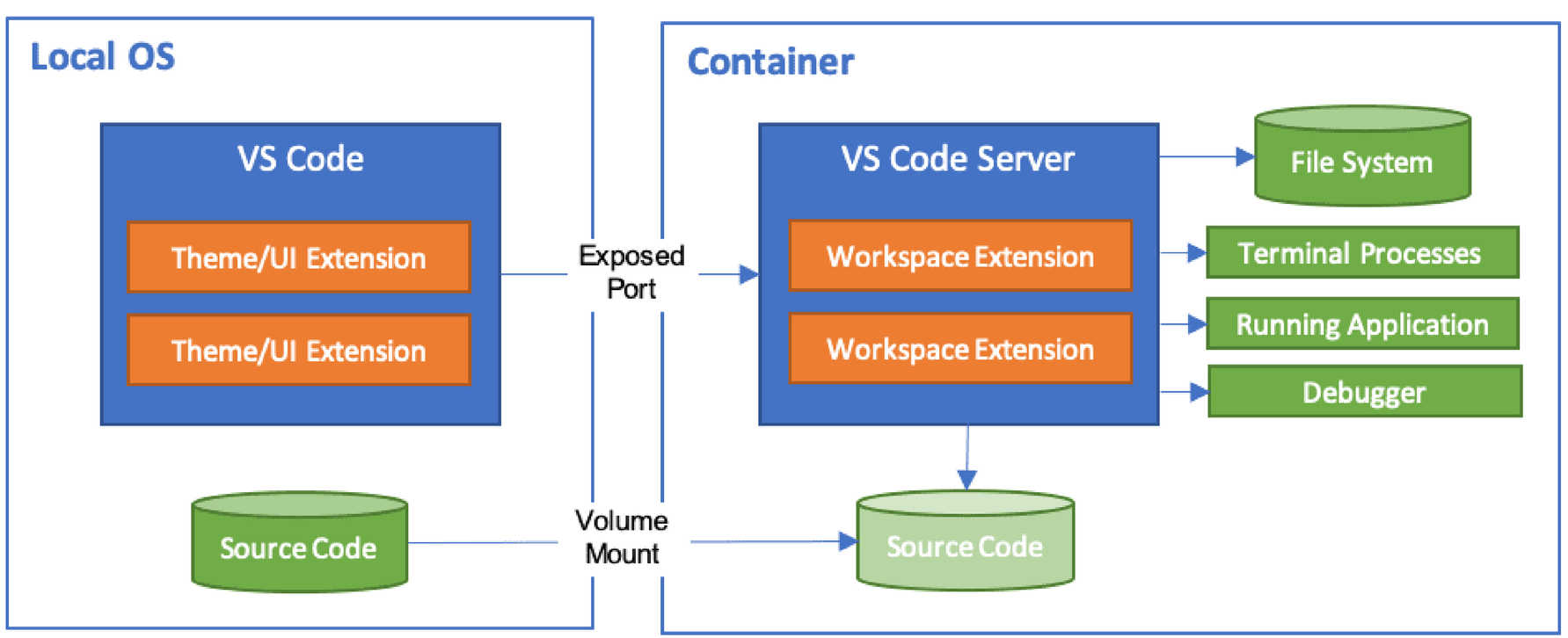
COSMO CONSULT

Business-Software for People

Prereq: VS Code dev container

What's that?

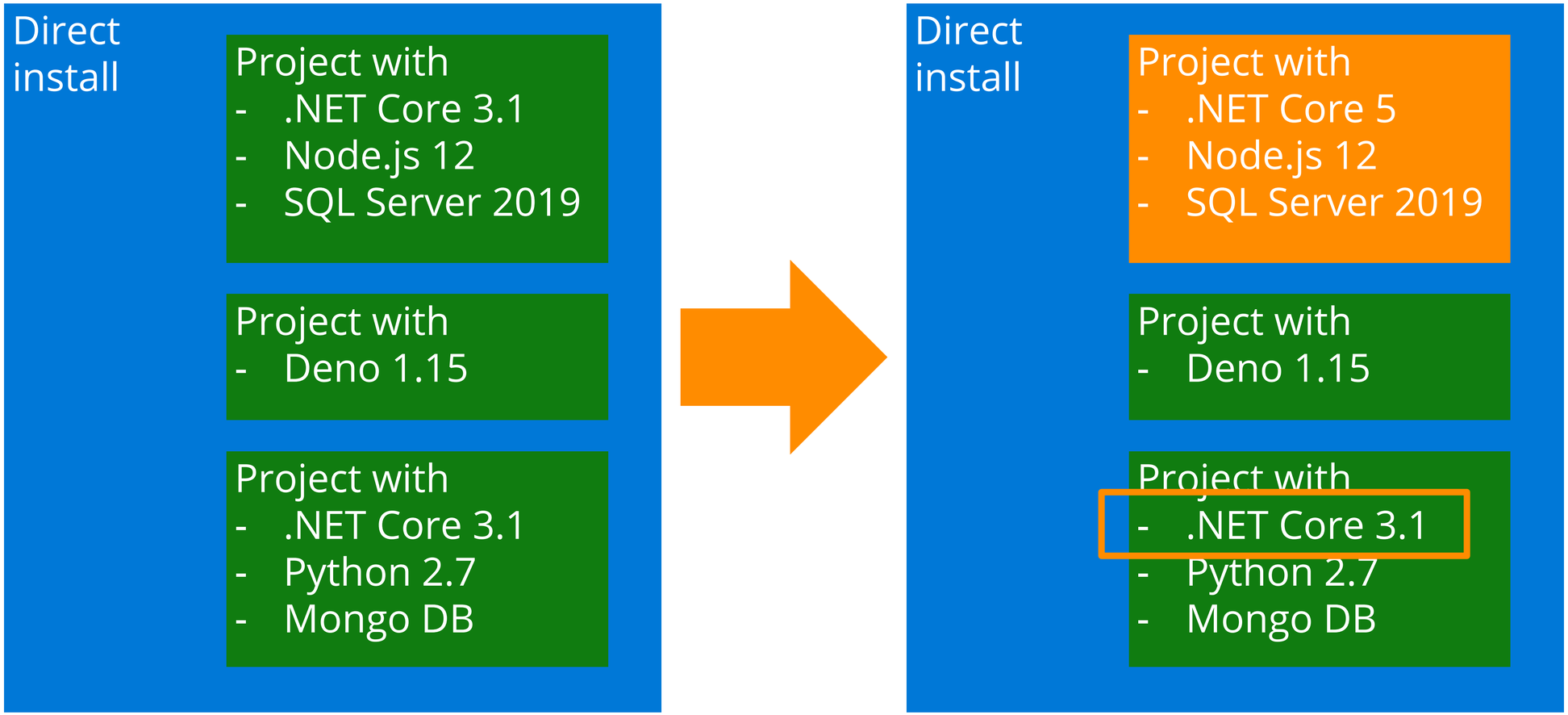
A mechanism to have easy to create, cleanly separated, standardized dev environments



Source: https://code.visualstudio.com/docs/remote/containers

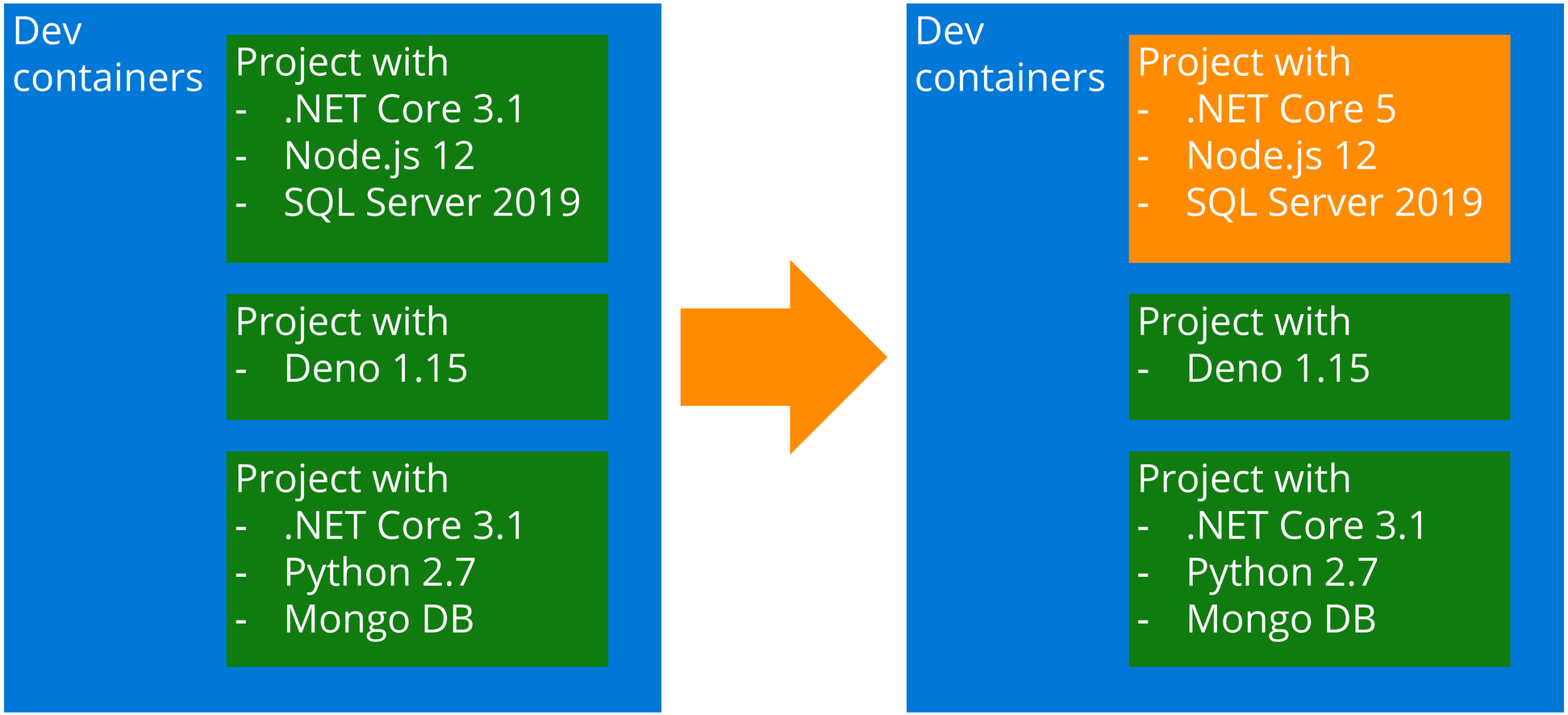
Prereq: VS Code dev container

What's that?



Prereq: VS Code dev container

What's that?



Prereq: VS Code dev container

What's that?

Configured through devcontainer.json

```
{
    "name": "C# (.NET)",
    "build": {
        "dockerfile": "Dockerfile",
        "args": {
            // Update 'VARIANT' to pick a .NET Core version: 2.1, 3.1, 5.0
            "VARIANT": "5.0",
            "INSTALL_NODE": "true",
            "NODE_VERSION": "12",
            "INSTALL_AZURE_CLI": "false"
        }
    },
    "extensions": [
        "ms-dotnettools.csharp",
        "eamodio.gitlens",
        "firefox-devtools.vscode-firefox-debug"
    ],
    ...
}
```

Prereq: VS Code dev container

What's that?

Set up through Dockerfile

```
ARG VARIANT="5.0"
FROM mcr.microsoft.com/vscode/devcontainers/dotnet:0-${VARIANT}

# [Option] Install Node.js
ARG INSTALL_NODE="true"
ARG NODE_VERSION="12"
RUN if [ "${INSTALL_NODE}" = "true" ]; then su vscode -c "umask 0002 && .
/usr/local/share/nvm/nvm.sh && nvm install ${NODE_VERSION} 2>&1"; fi

# [Option] Install Azure CLI
ARG INSTALL_AZURE_CLI="false"
COPY library-scripts/*.sh library-scripts/*.env /tmp/library-scripts/
RUN if [ "$INSTALL_AZURE_CLI" = "true" ]; then bash /tmp/library-scripts/azcli-debian.sh; fi \
    && apt-get clean -y && rm -rf /var/lib/apt/lists/* /tmp/library-scripts

# [Optional] Uncomment this section to install additional OS packages.
# RUN apt-get update && export DEBIAN_FRONTEND=noninteractive && apt-get -y install …

# [Optional] Uncomment this line to install global node packages.
RUN su vscode -c "source /usr/local/share/nvm/nvm.sh && npm install -g webpack webpack-cli tfx-
cli webpack-dev-server" 2>&1
```

Prereq: VS Code dev container

What's that?

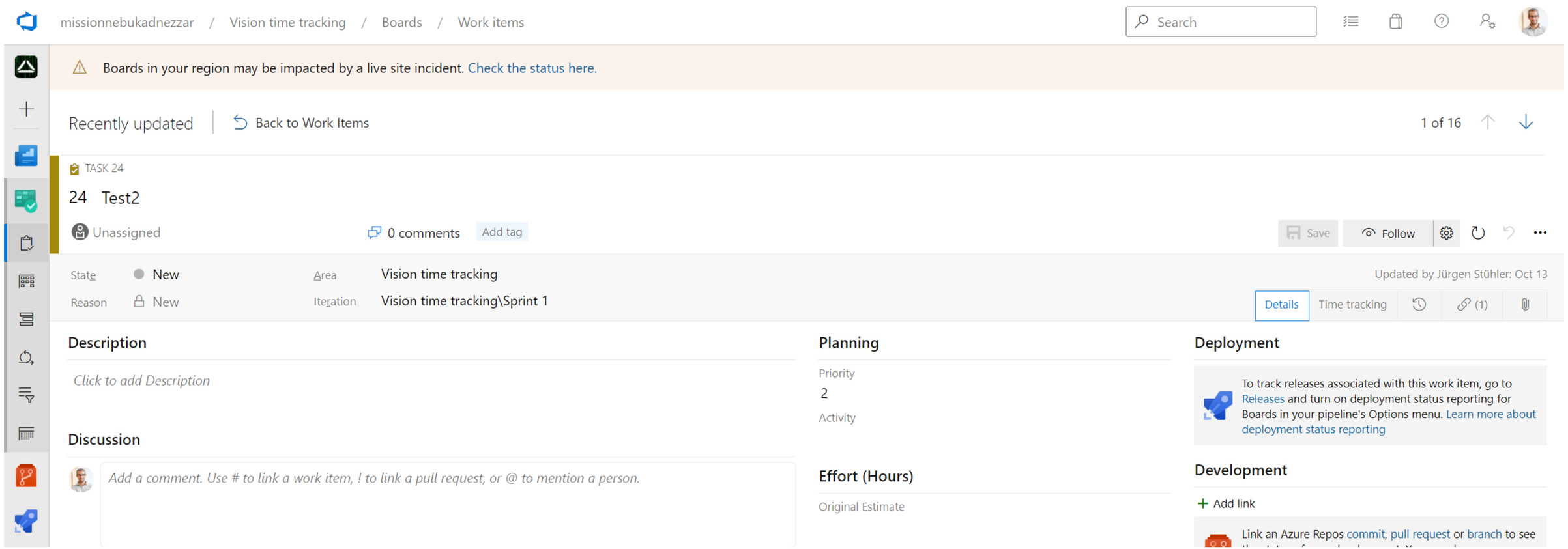
- Needs Windows Subsystem for Linux (WSL2)
  - Very easy setup through Windows Store
  - Container management through Docker Desktop ("Switch to Linux containers")
- Needs VS Code "Remote – Containers" extension
  - Might as well install "Remote Development" extension pack including "Remote – SSH" (with Windows support) and "Remote – WSL"
- Run "Remote-containers: Clone repository in Container volume" action

Prereq: VS Code dev container

What's that?

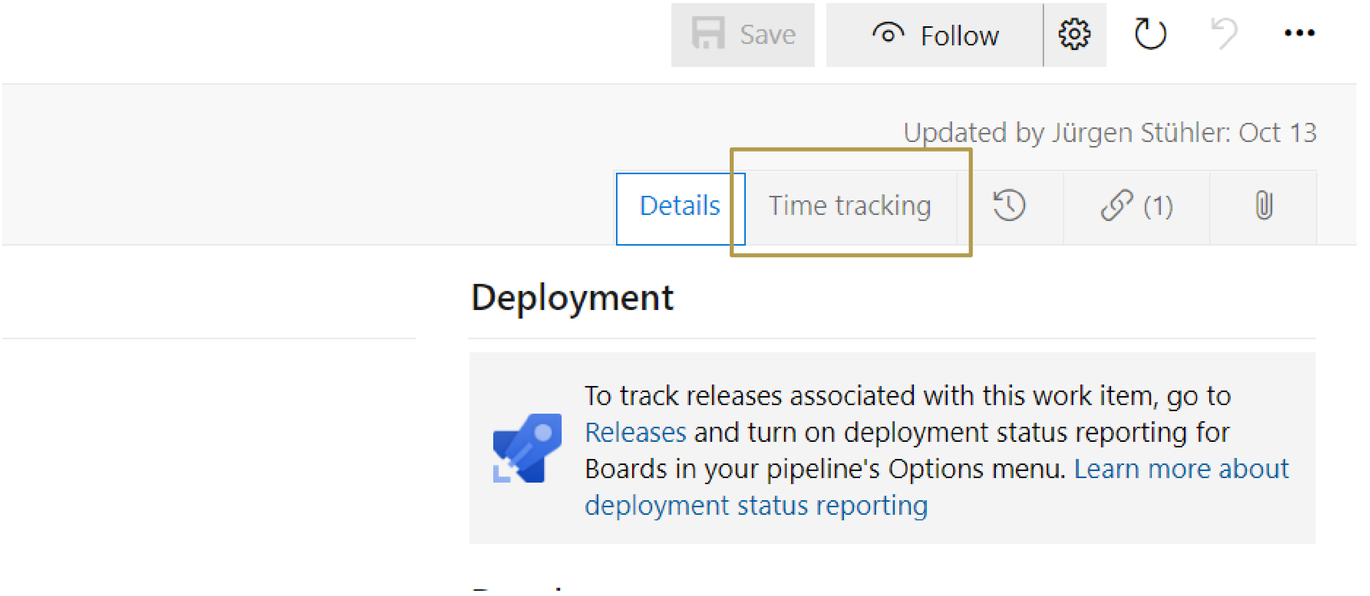
Azure DevOps Web Extension

What's that?



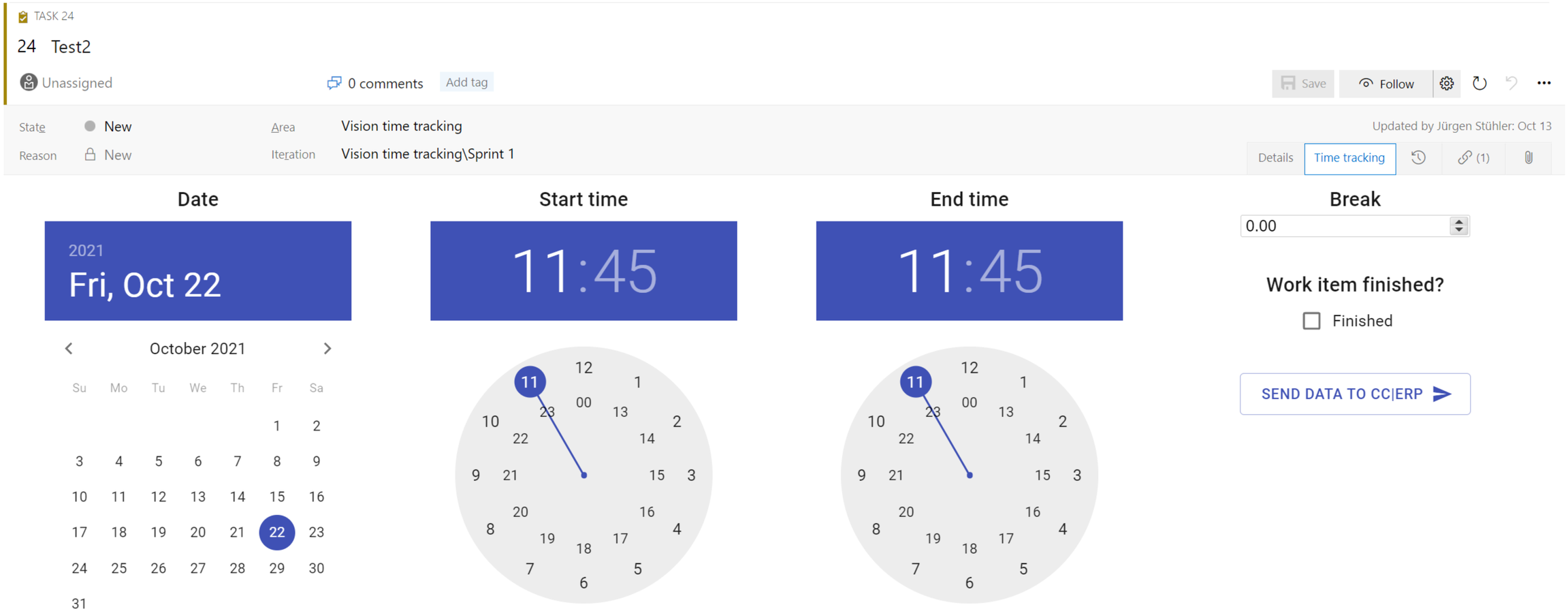
Azure DevOps Web Extension

What's that?



Azure DevOps Web Extension

What's that?



Azure DevOps Web Extension

How does it work?

Configured through manifest file vss-extension.json (1/3)

```
{
  "manifestVersion": 1,
  "id": "<put-your-publisher-here>-example-dev",
  "public": false,
  "baseUri": "https://localhost:3000",
  "version": "0.0.1",
  "name": "Dev Extension",
  "description": "Extension for development tests",
  "publisher": "<put-your-publisher-here>",
  "demands": [
    "api-version/3.0"
  ],
  "categories": [
    "Azure Boards"
  ],
  "targets": [
    {
      "id": "Microsoft.VisualStudio.Services"
    }
  ],
```

How does it work?

Configured through manifest file vss-extension.json (2/3)

```
"content": {
  "details": {
    "path": "README.md"
  }
},
"icons": {
  "default": "img/world.png"
},
"files": [
  {
    "path": "dist",
    "addressable": true
  },
  {
    "path": "img",
    "addressable": true
  }
],
```

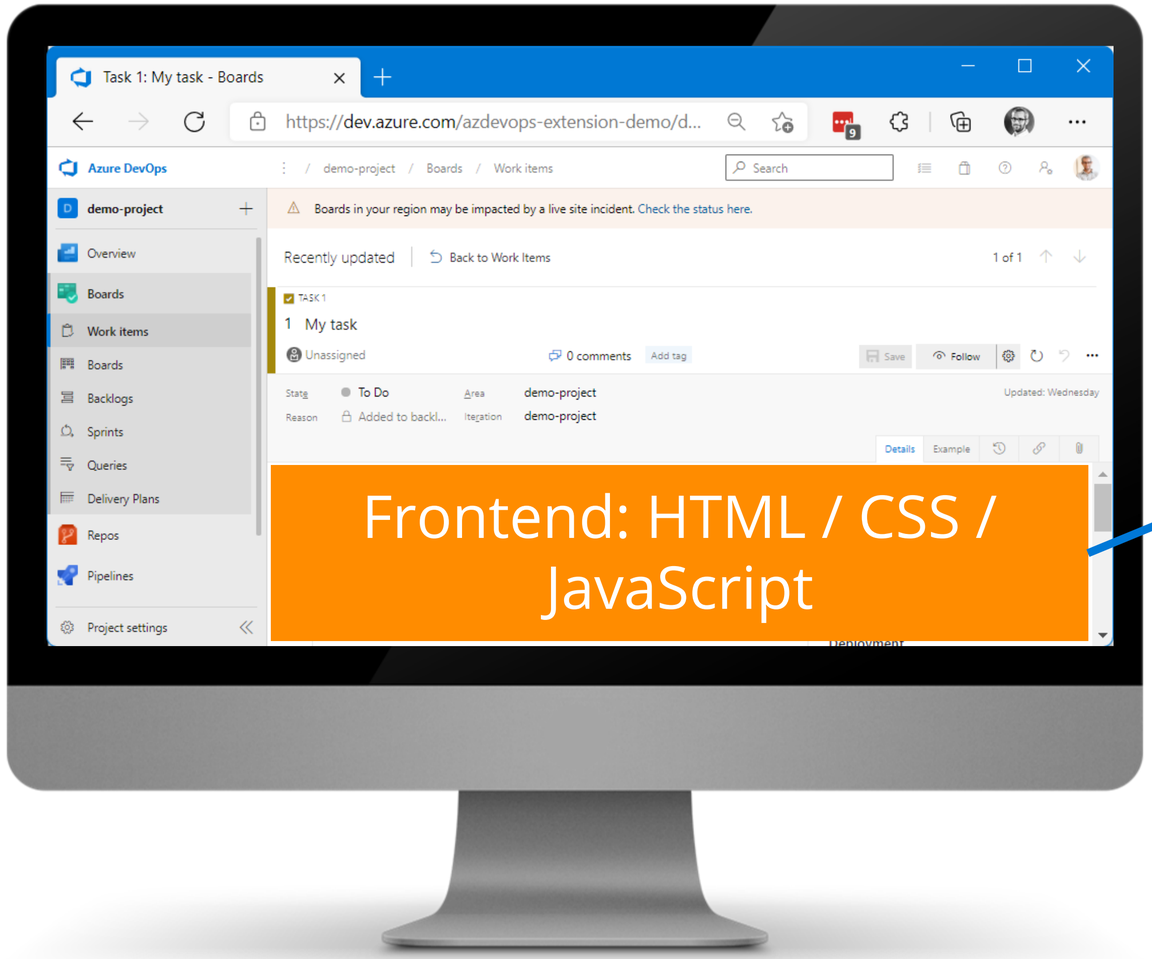
How does it work?

Configured through manifest file vss-extension.json (3/3)

```
  "contributions": [
    {
      "id": "example-work-item-form-page",
      "type": "ms.vss-work-web.work-item-form-page",
      "description": "Example page",
      "targets": [
        "ms.vss-work-web.work-item-form"
      ],
      "properties": {
        "name": "Example",
        "uri": "dist/example/example.html"
      }
    }
  ]
}
```

Azure DevOps Web Extension

How does it work?

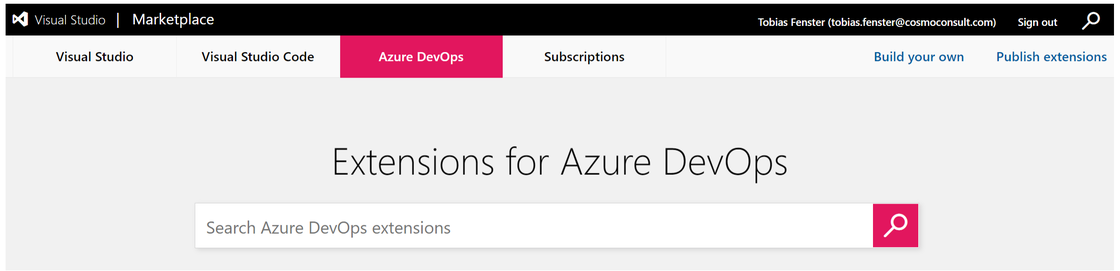


Backend: .NET Core Web API

Azure DevOps Web Extension

How does it work?

Build & Publish



Install

Extension (Manifest)

:3000

Backend: .NET Core Web API

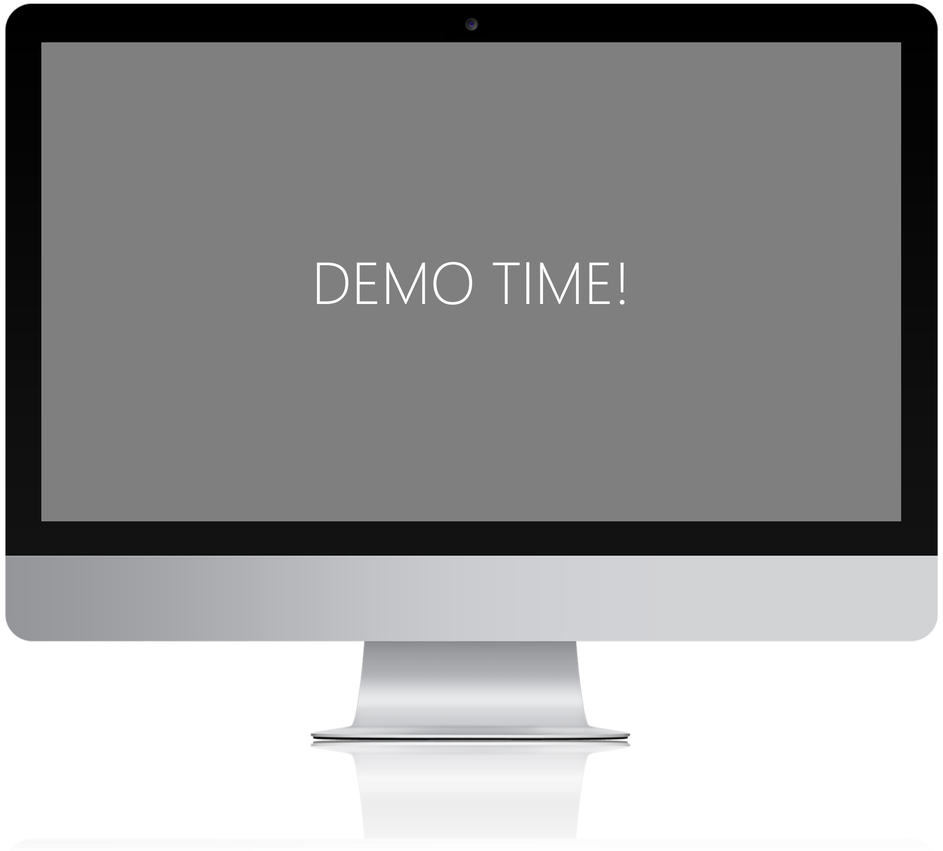
Dev Container

:5000

Frontend: HTML / CSS / JavaScript

Azure DevOps Web Extension

Let’s try!



Thanks for your attention

Any questions?

Tobias Fenster
CTO COSMO CONSULT

DIRECTIONS
4 PARTNERS

www.directions4partners.com