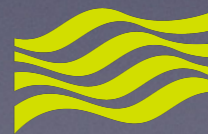


# OpenWIS v4

*Development approach based on overlays*



**Met Office**

Toulouse, 20-23 March 2017

# GitHub developer's workflow

---

Find the project you want to contribute to

Fork it in your private repo

Make changes, push changes to private repo,  
Pull-Request

When PR is accepted, delete your local fork/branch

Develop a new feature



# GitHub developer's workflow

---

Works well for "small" features

Assumes a good PR-acceptance turnaround time

Assumes upstream project is interested for all types of contributions

"Bound" to the technical choices of the upstream project

Difficult for derivate, independent work

# Fork-based approach

## Fork-based development process

Initial release process works smoothly as it only contains modifications to an existing version/tag



Subsequent release merges may face issues due to:

- OpenWIS fork not kept up-to-date
- Duplication in feature implementation
- Technology divergence
- Changes in CGN may require major effort in OpenWIS

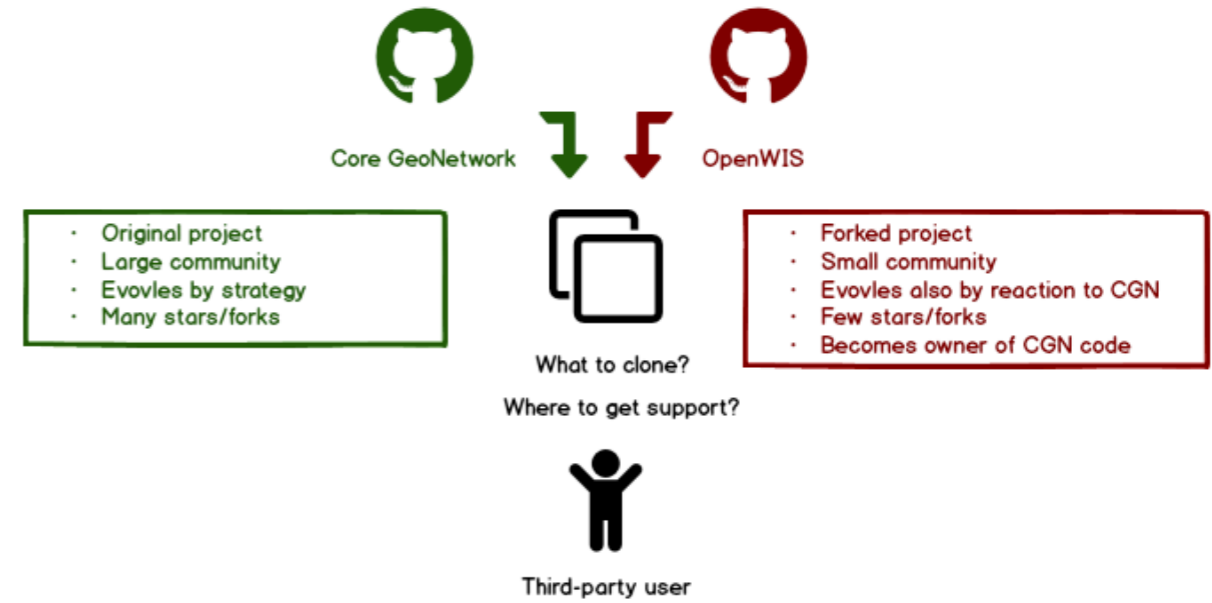


New risk is introduced in the form of:

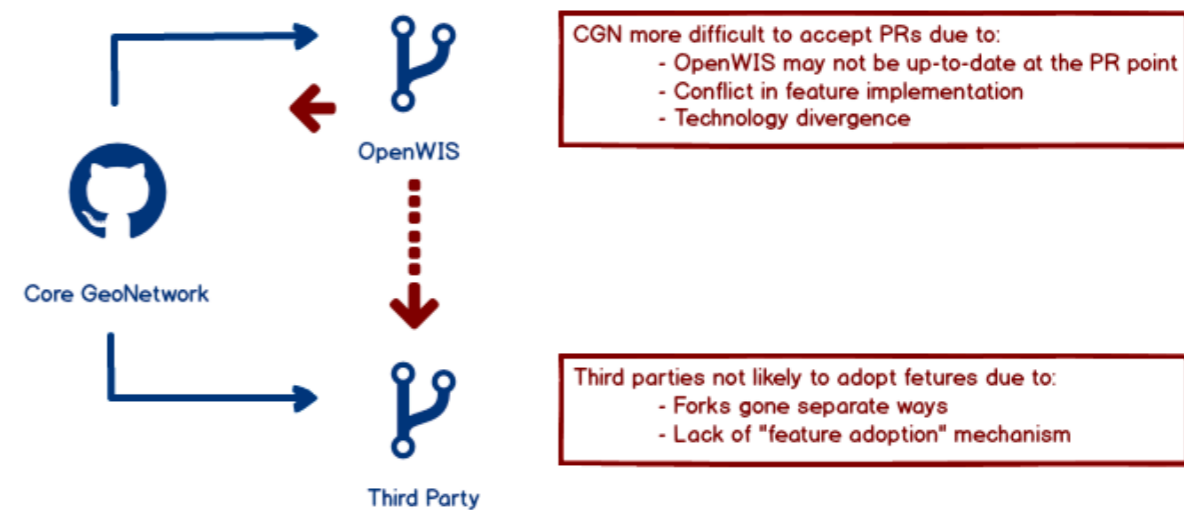
- OpenWIS features must follow CGN technology
- Long-term OpenWIS planning not possible without knowledge of CGN plans



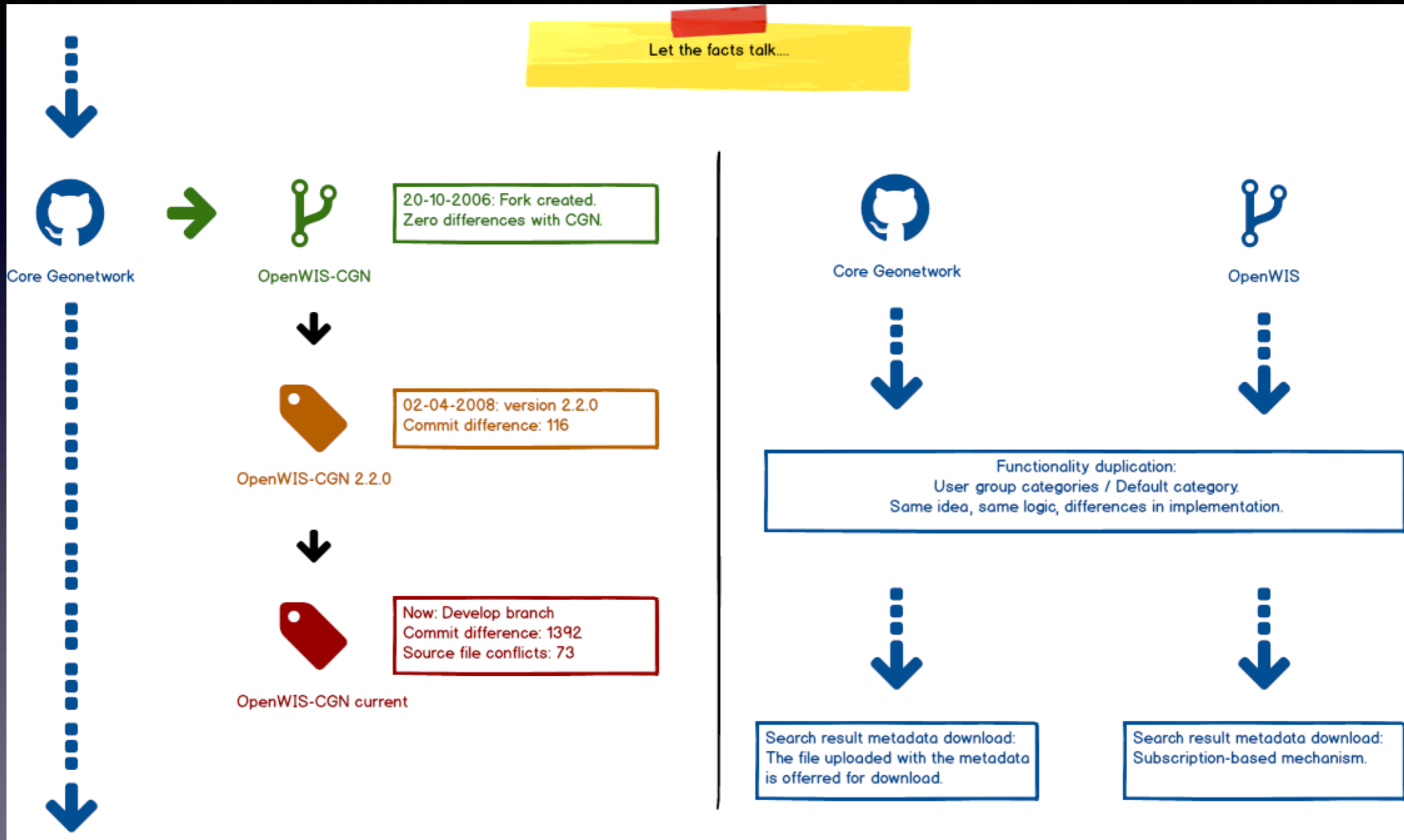
## Third-party usage



## Contributing to the community



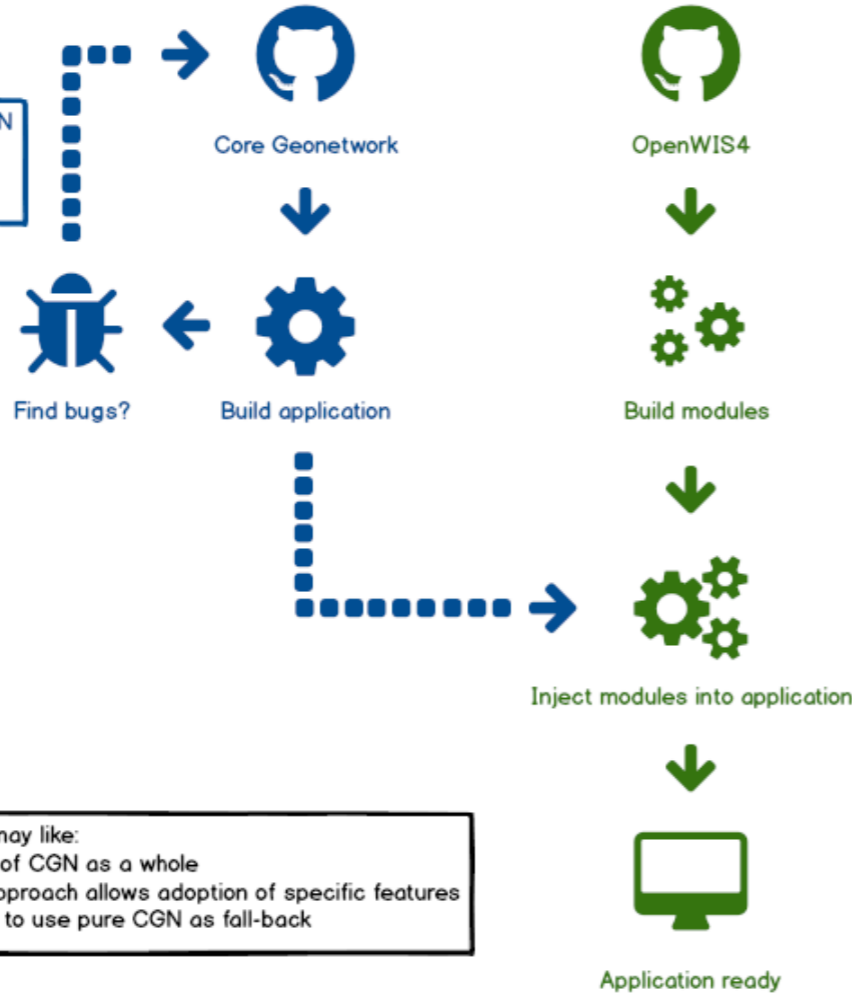
# Fork-based problems in practice



# An overlay-based approach

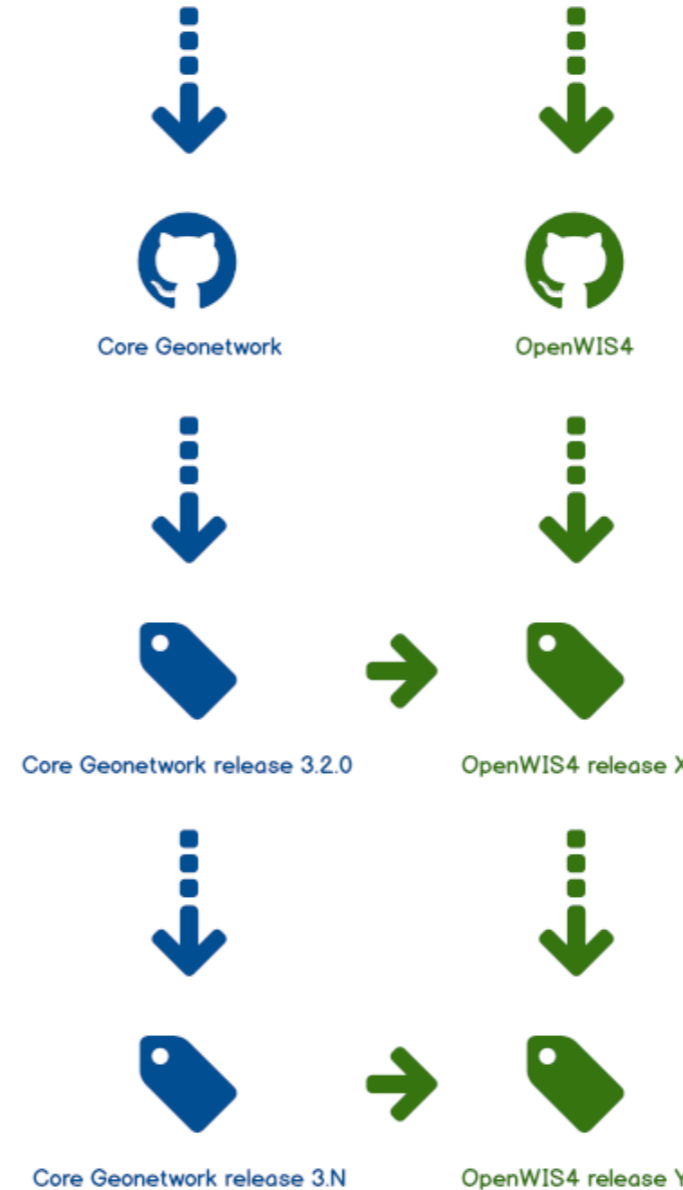
## The v4 approach

PR on the original code of CGN gives higher chance of acceptance



Third-party users may like:

- Inclusion of CGN as a whole
- Plug-in approach allows adoption of specific features
- Flexibility to use pure CGN as fall-back



...  
`<cgcn.version>3.2.0-0</cgcn.version>`  
 ...

...  
`<cgcn.version>3.N</cgcn.version>`  
 ...



The downside of the v4 approach:

- Not for beginners
- Local build of CGN artifacts
- Plug-in approach requires extra effort for each feature

# So, does it work?

Initial integration was with CGN 3.2.0

At 10-Feb 2017, CGN 3.2.1 was released

The screenshot displays the 'Settings' page in the GeoNetwork administration console. The page is titled 'System settings' and is divided into three main sections: 'Catalog description', 'Catalog', and 'Catalog server'. The 'Catalog description' section includes fields for 'Catalog name' (My GeoNetwork catalogue), 'Catalogue identifier' (a7eb06c0-b482-423d-b500-a34faaa729aa), 'Organization' (My organization), and 'SVN UUID' (60c08e9d-a9f4-47bc-b521-3846cc38b695). The 'Catalog' section includes 'Version' (3.2.1) and 'Minor version' (0). The 'Catalog server' section includes 'Host' (localhost). The page also features a navigation menu at the top with options like 'Settings', 'Logo', 'Sources', 'CSW', 'Virtual CSW', 'CSW test', and 'Map servers'. A sidebar on the right contains a list of settings categories, including 'Catalog description', 'Catalog', 'Catalog server', 'Intranet', 'Proxy server', 'system/cors', 'Feedback', 'Metadata Search Results', 'Catalog Service for the Web (CSW)', 'User self-registration', 'User feedback', 'Link in metadata records', 'Metadata rating', 'Download service', 'Metadata XLink', 'Metadata / ISO19139 / Nil reason attribute with held', 'Metadata update', 'Search statistics', 'Index optimization', 'Open Archive Initiative (OA-PMH) Provider', 'INSPIRE Directive configuration', 'Harvesters', 'Harvester', 'Metadata create', 'Metadata configuration', 'Metadata privileges', 'Indexing', 'Language detection', and 'Search & language'.

# So, does it work?

---

## Maven changes

```
<openwis.cgn.version>3.2.0-0</openwis.cgn.version>
```

=>

```
<openwis.cgn.version>3.2.1-0</openwis.cgn.version>
```

## Additional changes

```
selectNoXLinkResolver(String, boolean)
```

=>

```
selectNoXLinkResolver(String, boolean, boolean)
```

Having OpenWIS based on CGN 3.2.1 actually took us...

...20 mins!