Andy Williams

Derek Reid

fyne conf 2022

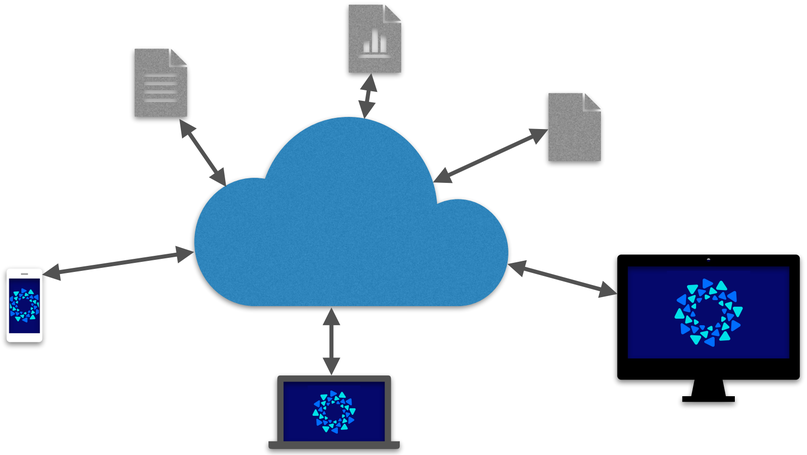
Upgrading your app with cloud sync

For data as universally available as your apps

Why Cloud?

fyne conf 2022

- Users should have access to their data whenever, wherever
- Fyne apps run on every device, but data is local
- Working remote, on the train or back in the office
- Cloud storage is expected for mobile ecosystem



- Provide users with the power of cloud, on all devices

Harness the Power of the Cloud

fyne conf 2022

- Persist user data to remote service
- Sync preferences and stored documents
- Standard providers available including Dropbox
- Custom provider for business specific cloud

- Fyne cloud storage is as easy as building your first GUI

Integrate a Cloud Provider

fyne conf 2022

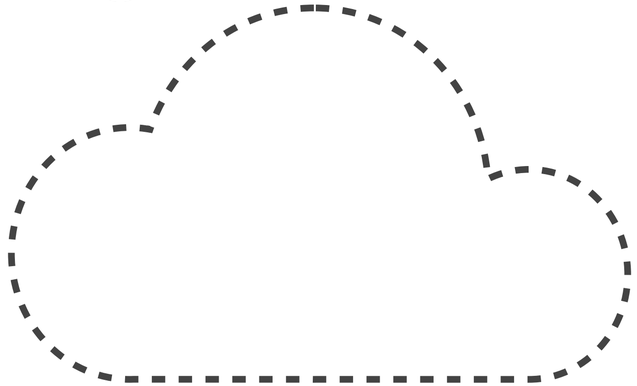
```
func main() {
    a := app.NewWithID("io.fyne.cloud.example")
    a.SetCloudProvider(mycloud.NewProvider())

    w := a.NewWindow("Cloud")
    ...
    w.ShowAndRun()
}
```

Implementing a Cloud Provider

fyne conf 2022

- Implement the `CloudProvider` interface
  - Perform any login or authentication actions in Setup method
- Export the type or constructor for passing to `SetCloudProvider()`
- Add the features of cloud functionality you will support
  - `CloudProviderStorage` to store documents
  - `CloudProviderPreferences` to sync user preferences



Implementing a Cloud Provider - Description

fyne conf 2022

```
// CloudProvider specifies the identifying information of a cloud provider.
type CloudProvider interface {
    // ProviderDescription returns a more detailed description.
    ProviderDescription() string
    // ProviderIcon returns an icon resource for the cloud service.
    ProviderIcon() Resource
    // ProviderName returns the name of this cloud service.
    ProviderName() string

    // Cleanup is called when this provider is no longer used.
   Cleanup(App)
    // Setup is called when this provider is being used for the first time.
   Setup(App) error
}
```

Implementing a Cloud Provider - Storage

fyne conf 2022

```go
// CloudProviderStorage interface defines the functionality that a cloud provider
// will include if it is capable of synchronizing user documents.
type CloudProviderStorage interface {
    // CloudStorage returns a storage provider that will sync documents to the cloud.
    CloudStorage(App) Storage
}
```

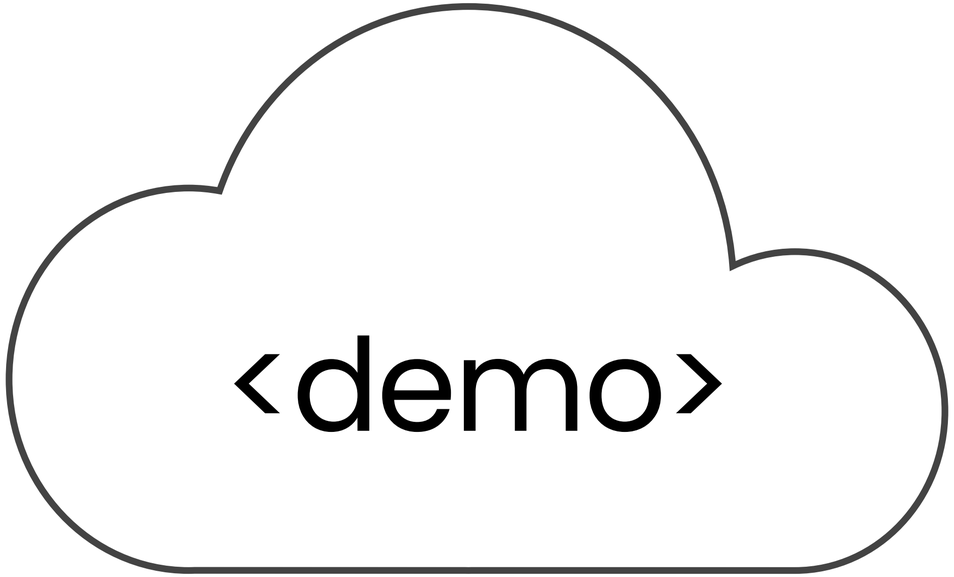
Implementing a Cloud Provider - Preferences

fyne conf 2022

```go
// CloudProviderPreferences interface defines the functionality that a cloud provider
// will include if it is capable of synchronizing user preferences.
type CloudProviderPreferences interface {
    // CloudPreferences returns a preference provider that will sync to the cloud.
    CloudPreferences(App) Preferences
}
```

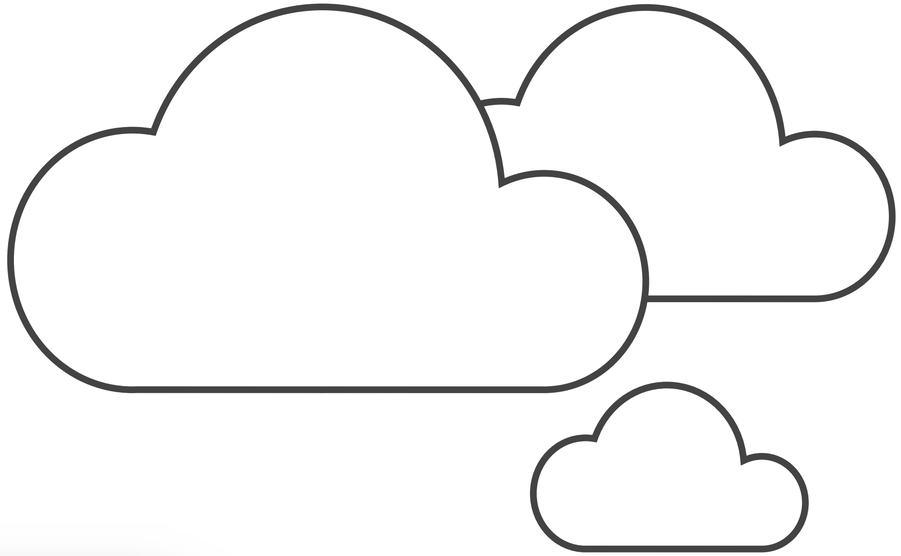
Implementing a Cloud Provider - Demo

fyne conf 2022



Add user selection of providers

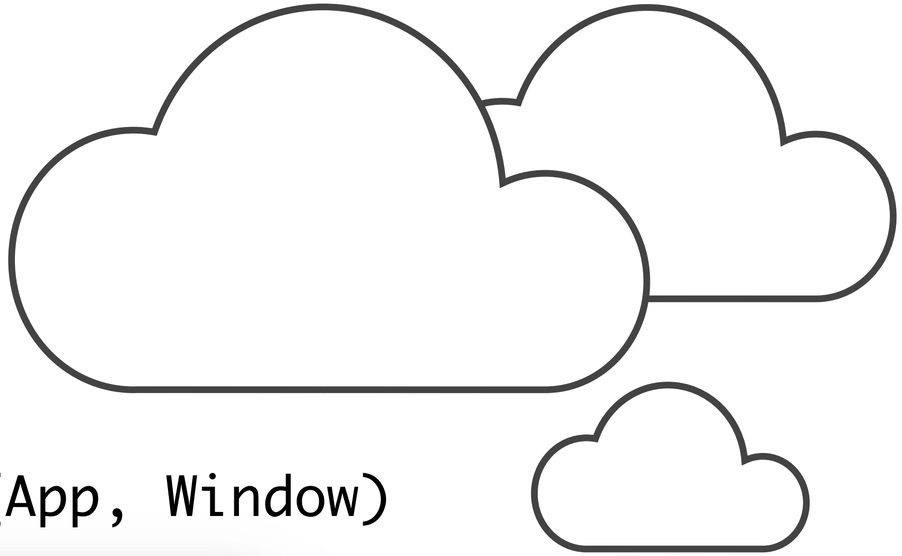
- Instead of hard coding a provider
- Offer user choice of data sync
- Stored globally for consistent cloud store
- Import “fyne.io/cloud” and call `cloud.Enable(App)`



```
func main() {
    a := app.NewWithID("io.fyne.cloud.example")
    cloud.Enable(a)

    w := a.NewWindow("Cloud")
    ...
    w.ShowAndRun()
}
```

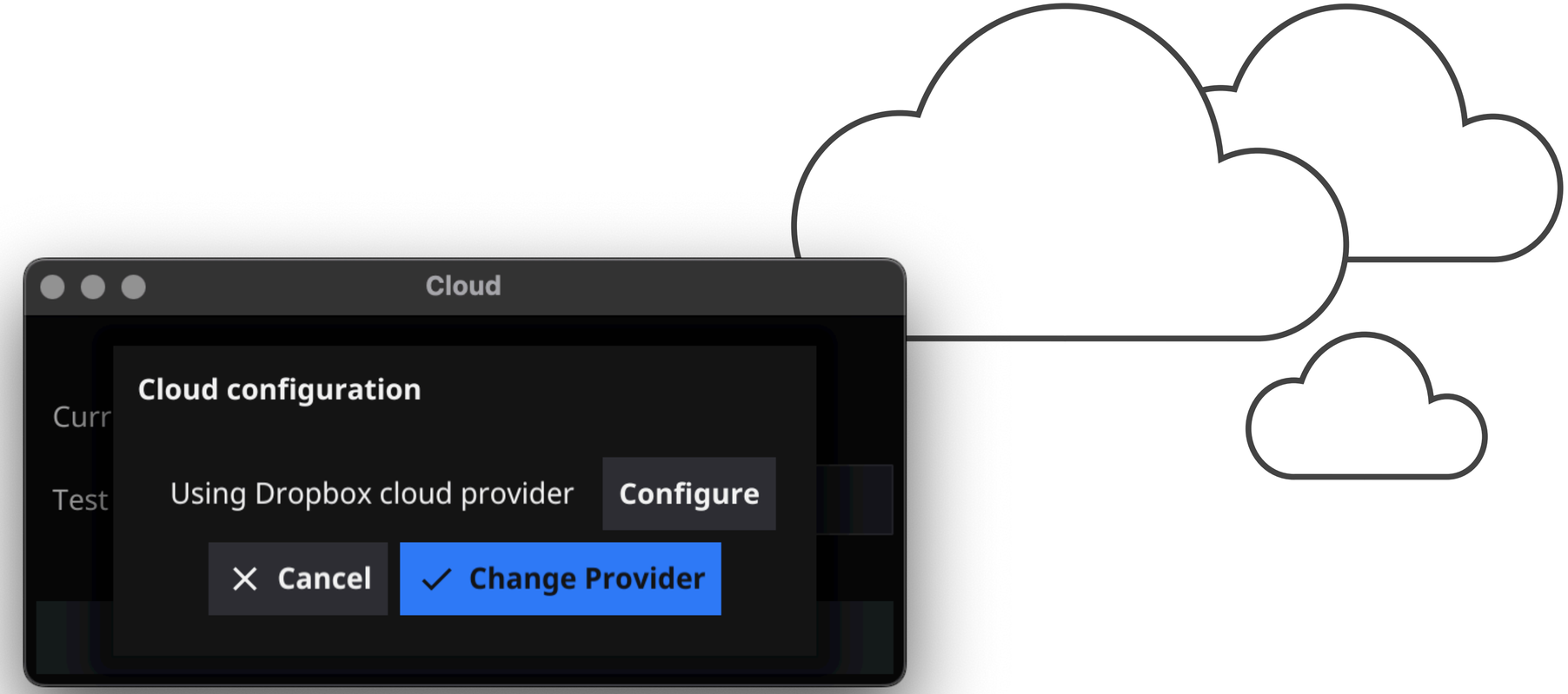
User Selection - Add Configuration



- Entry point for user choosing sync provider
- Opens user interface to select and configure
- From a menu or button just call `cloud.ShowSettings(App, Window)`

```
w := a.NewWindow("Cloud Notes")
w.SetMainMenu(fyne.NewMainMenu(
    fyne.NewMenu("File",
        fyne.NewMenuItem("Sync...", func() {
            cloud.ShowSettings(a, w)
        }))))
```

User Selection - Add Configuration



Cloud Sync for Fyne Apps

fyne conf 2022

- Ensure user data is always available
- Simple setup, easy to use
- Same Preference and Storage APIs
- Pick a provider, or allow user to choose
- Synced data, happy users