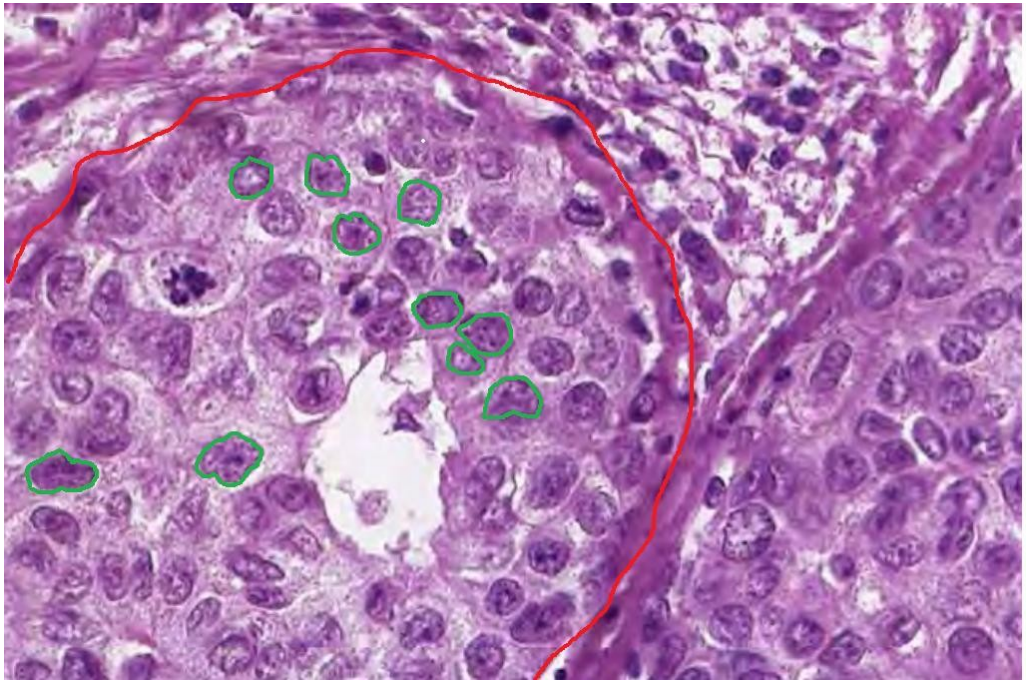
Exemple d’une application de cytologie

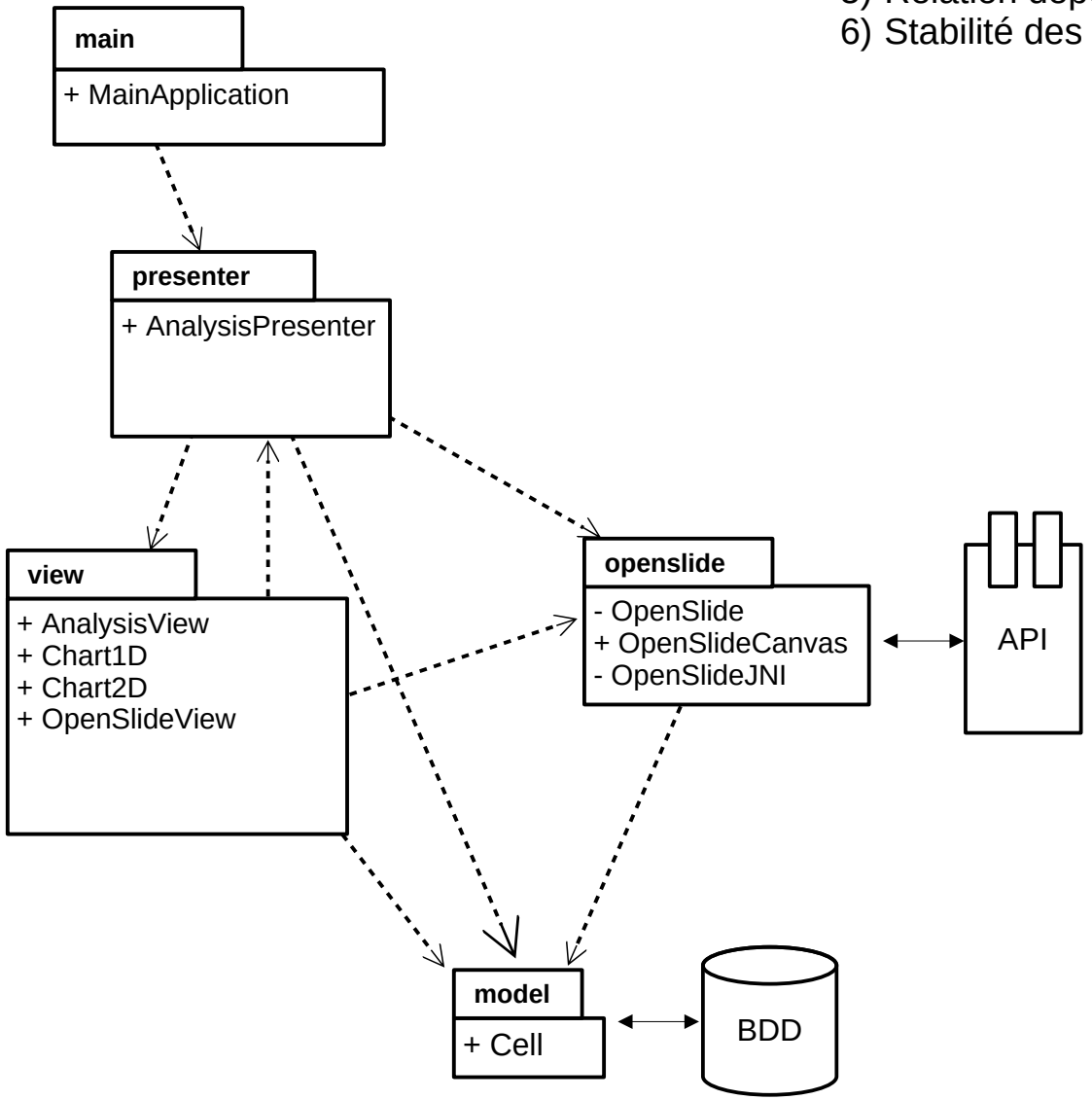
Le programme consiste à afficher aux médecins des statistiques concernant des caractéristiques de cellules segmentées dans des images de cytologie (caractéristiques topologiques, morphologiques, colorimétriques, photométriques…) en vue d’un dépistage précoce du cancer.

Le programme utilise la bibliothèque Openslide qui permet de lire et d’afficher des images médicales de grande taille (100 000 x 100 000 pixels).



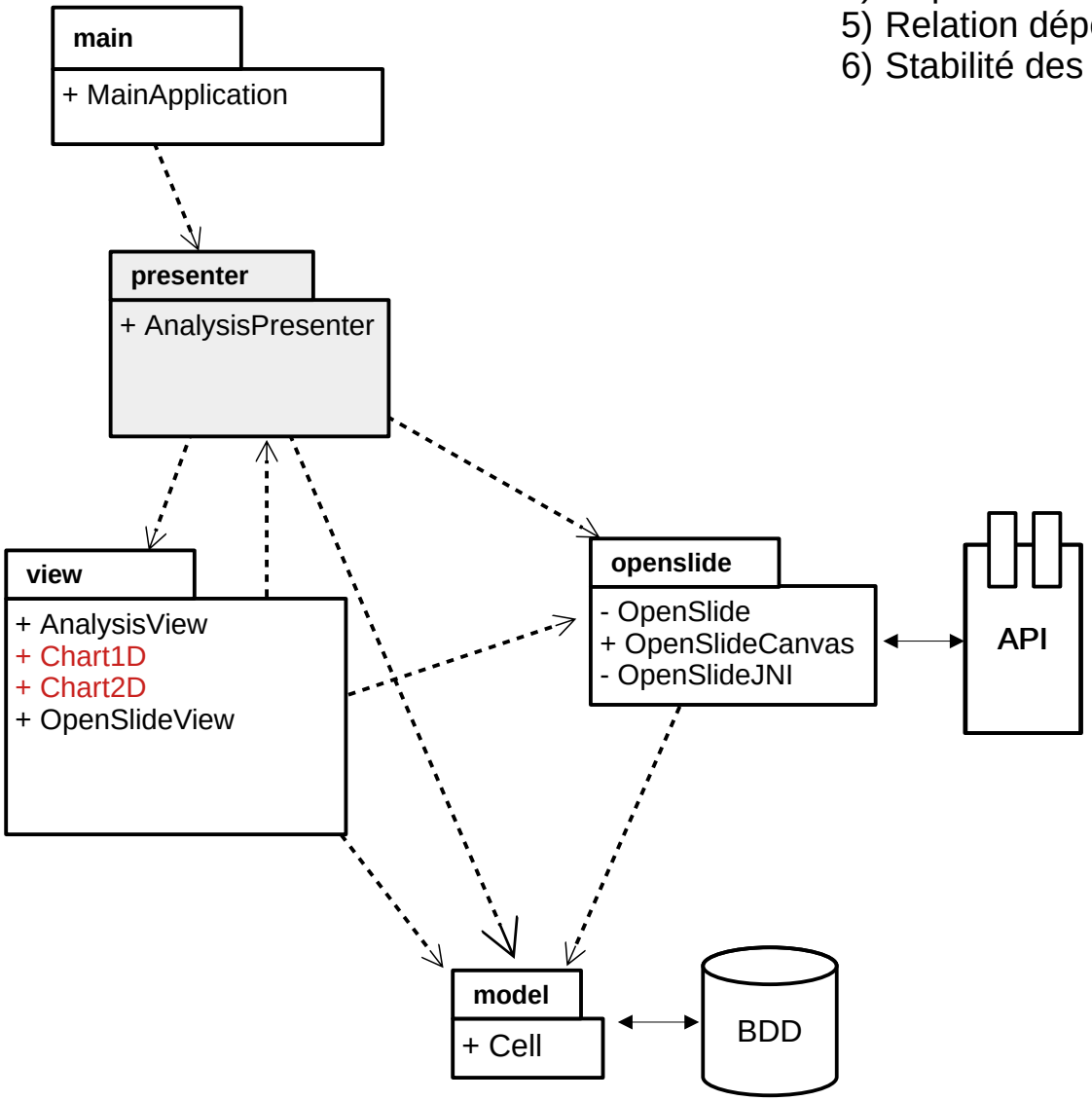
Soit le diagramme de paquets d'un projet qui reflète une architecture MVP.

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



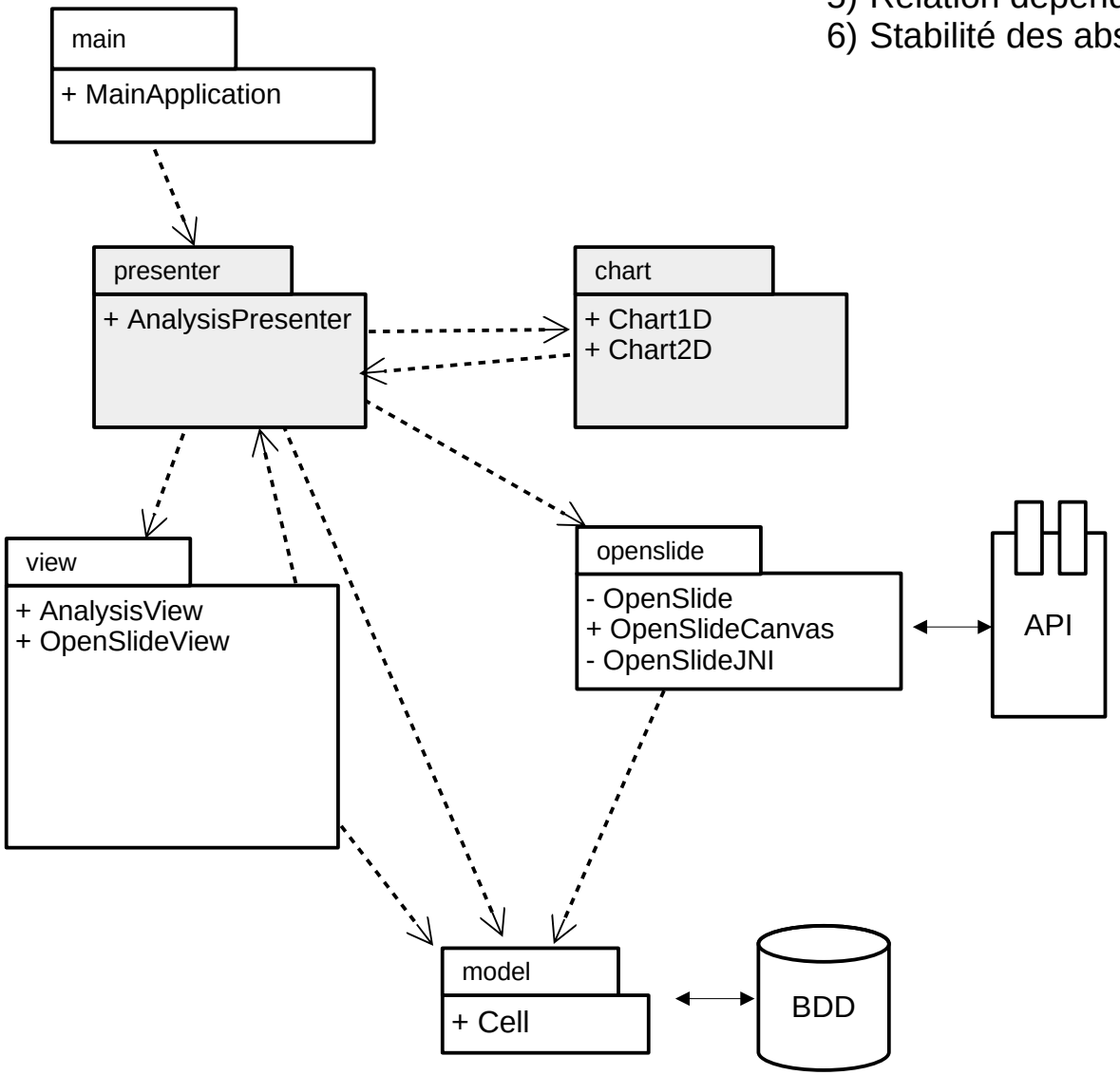
Problème : non respect de l'unicité de responsabilité dans le paquet view : analyse + chart

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



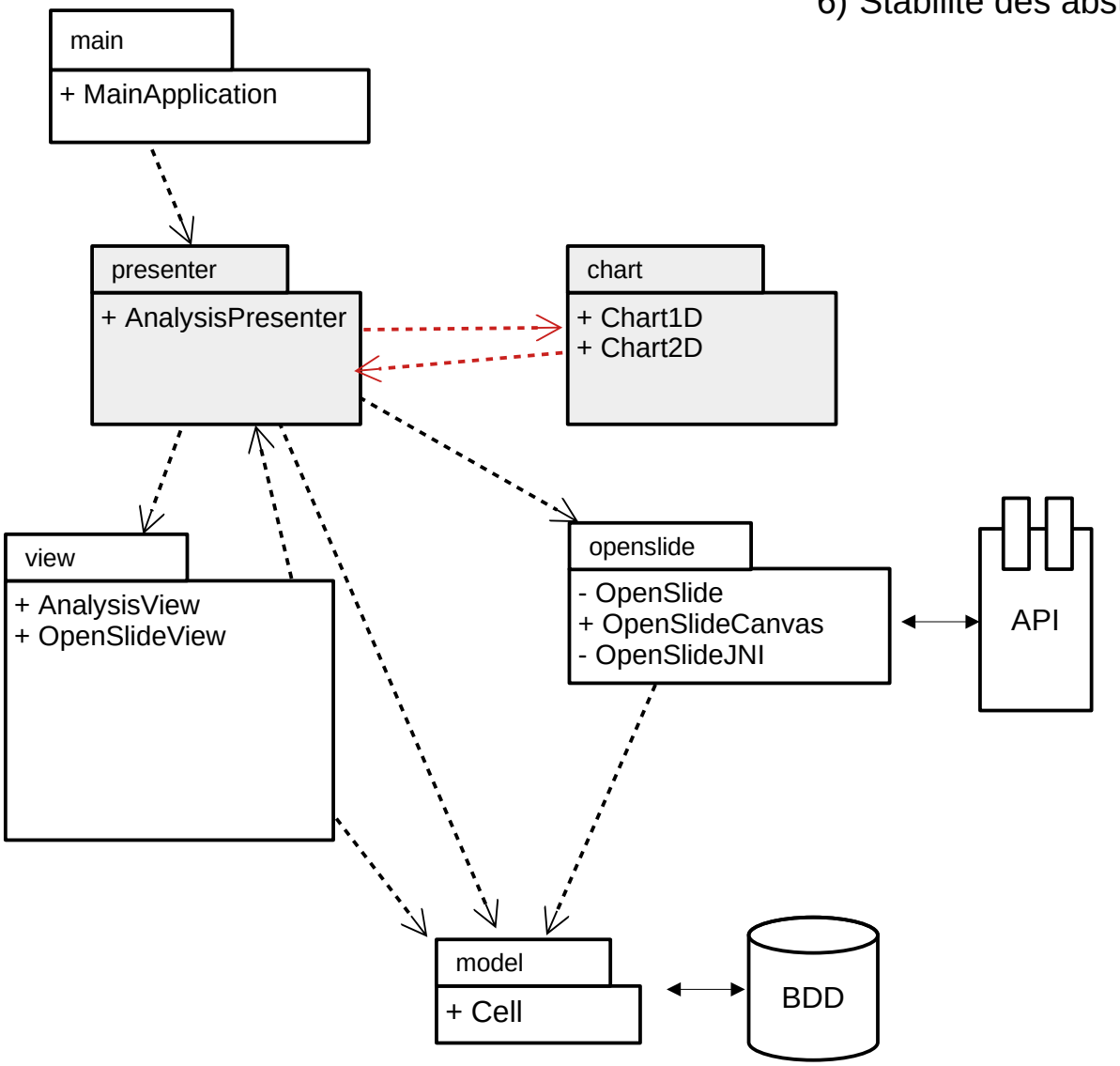
Solution : un paquet pour les charts

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



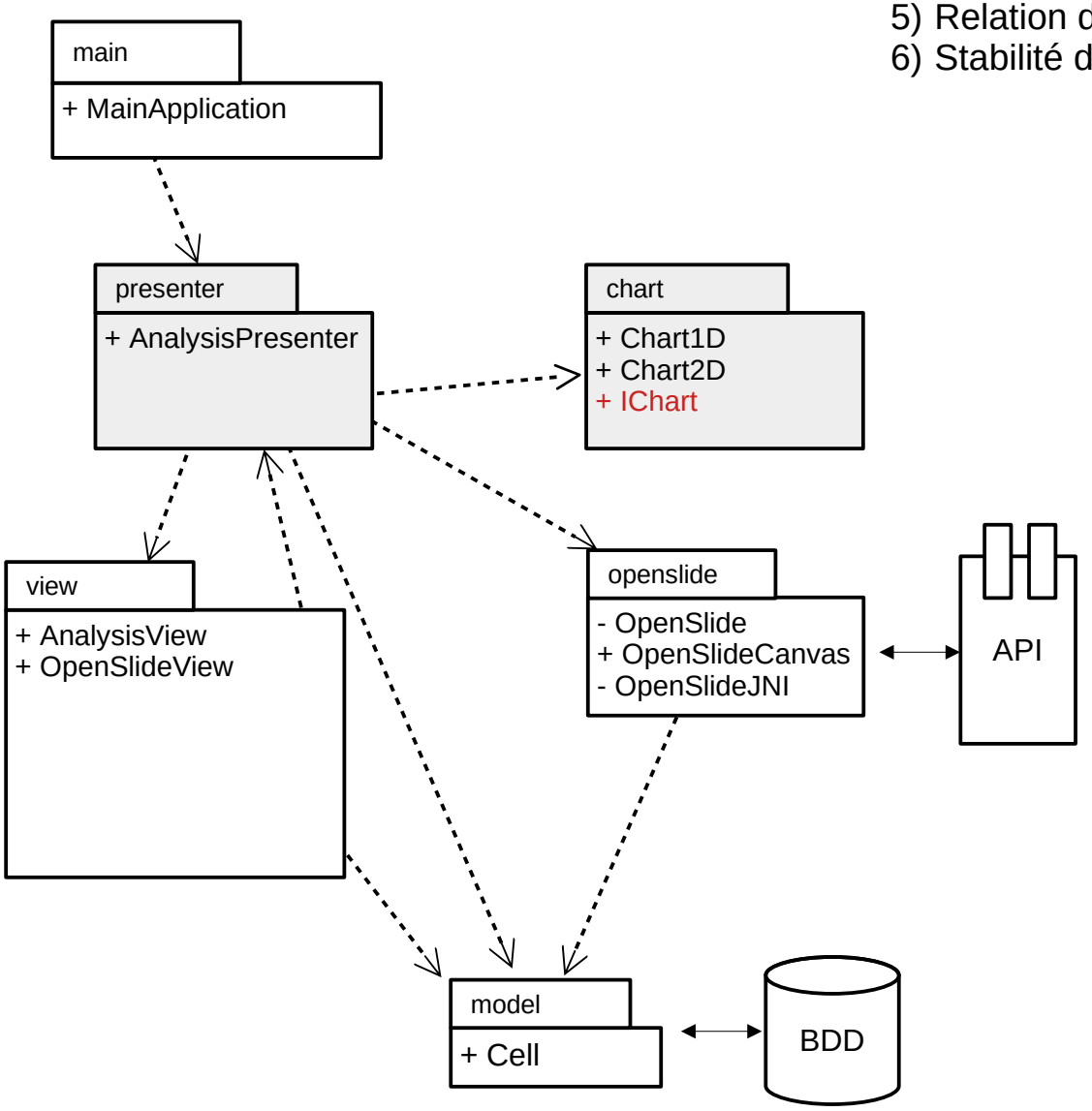
Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions

Problème : dépendance acyclique entre presenter et chart



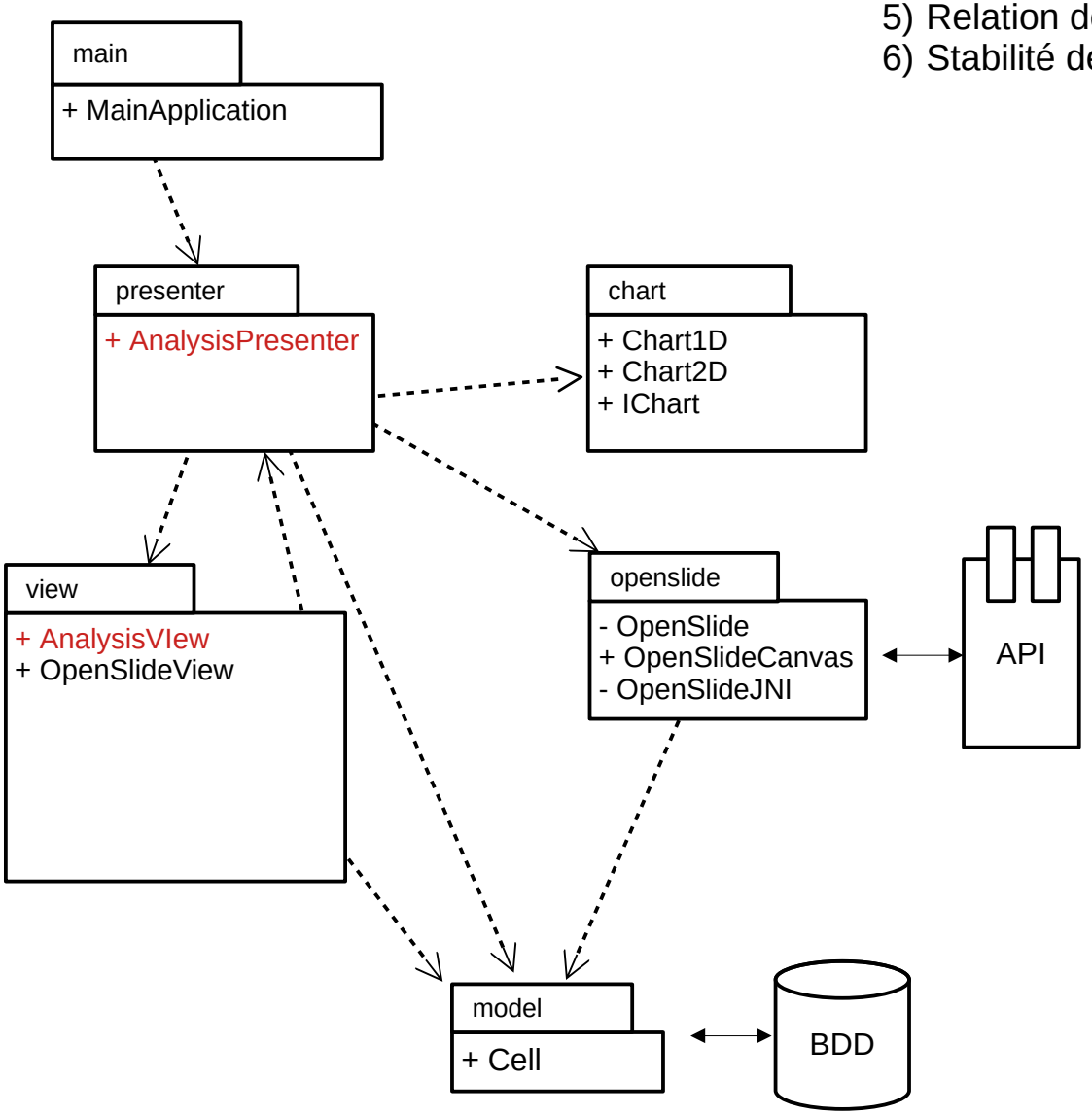
Solution : Ajouter une interface pour les charts. AnalysisPresenter construit les charts et passe les données en flux poussé (il ne passe pas les Cell mais une structure de donnes commune.

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



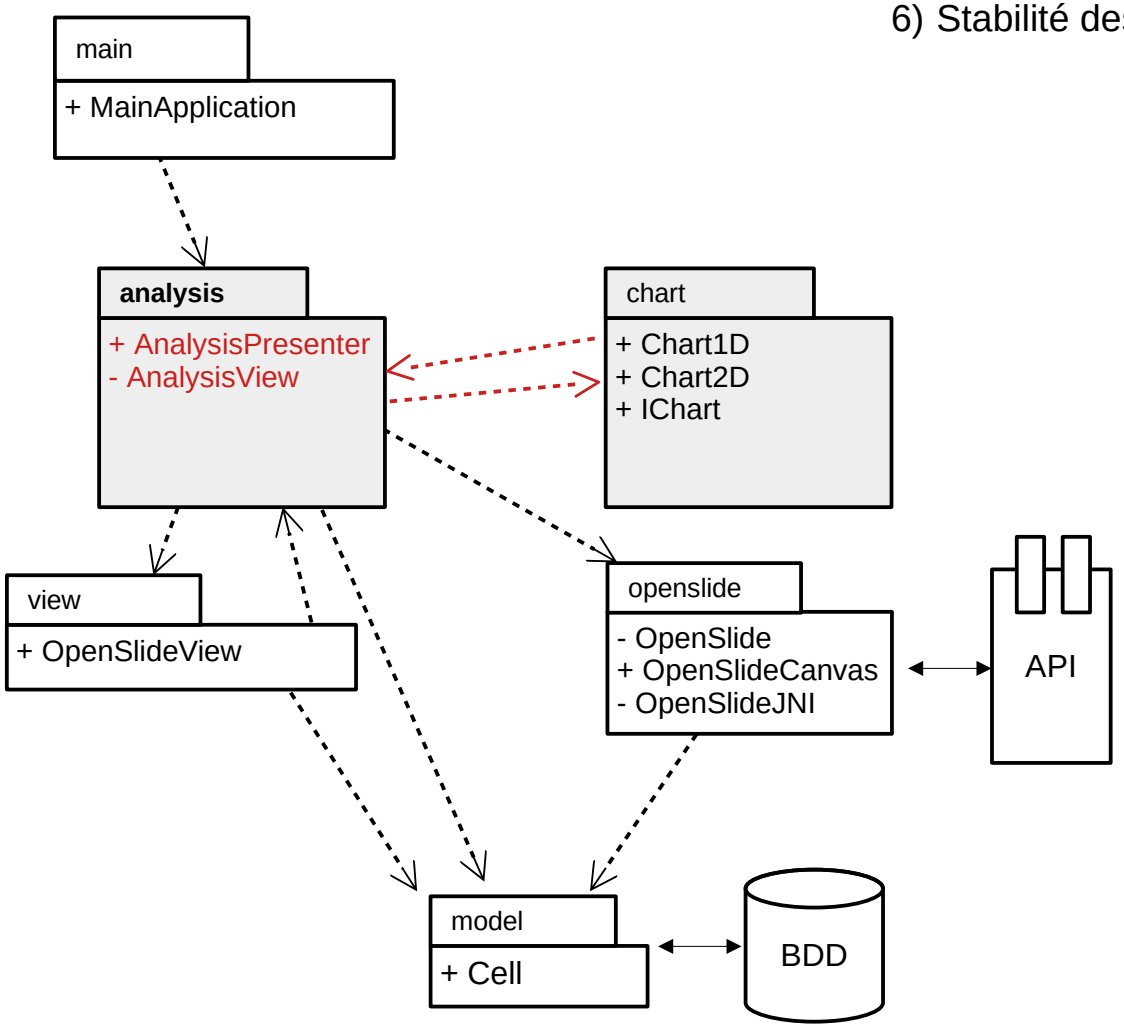
Problème : non respect de la fermeture et réutilisation commune entre les présentations et vue de analysis et chart

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



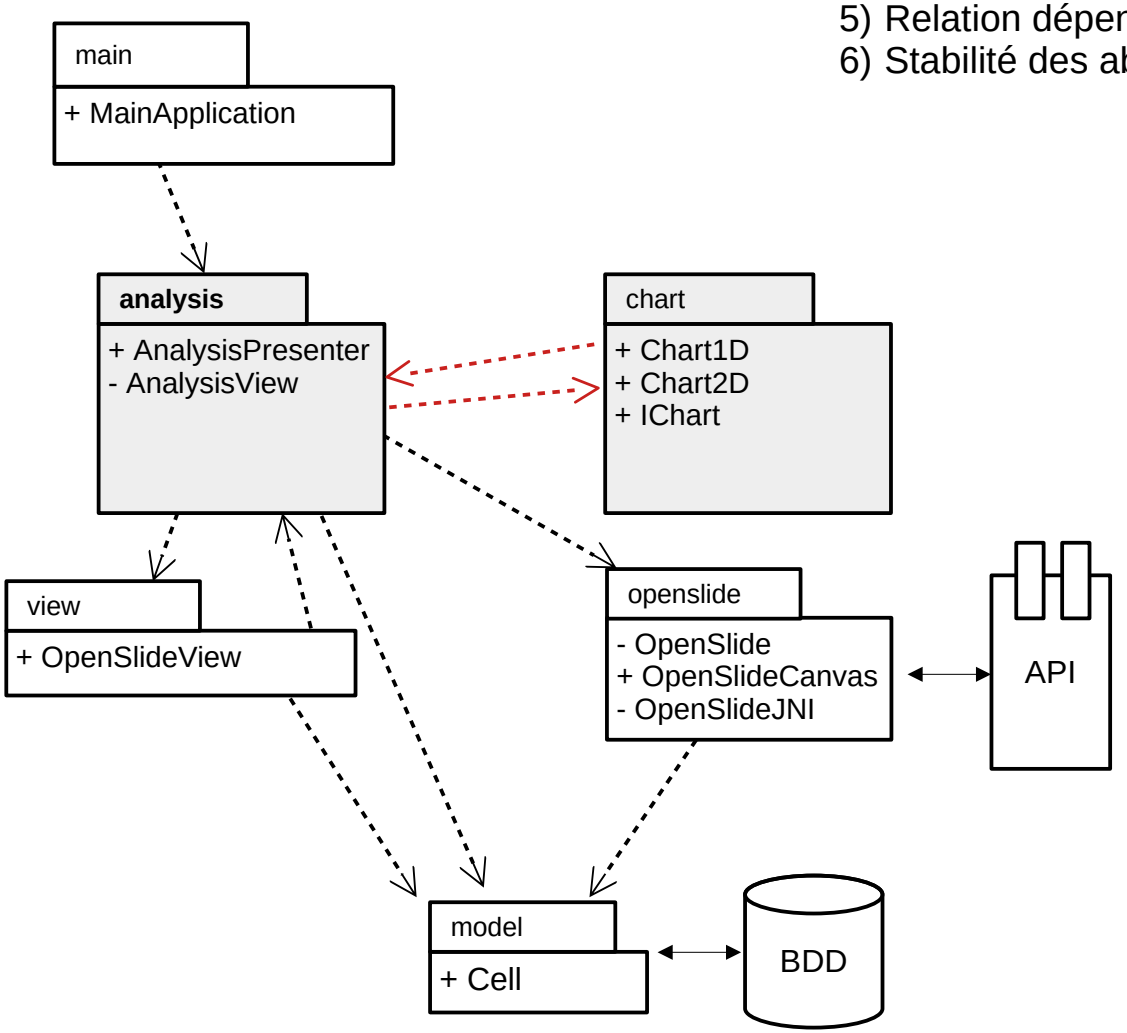
Solution : regrouper le présentateur et la vue dans le même paquet. Renommer le paquet presenter en analysis.

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



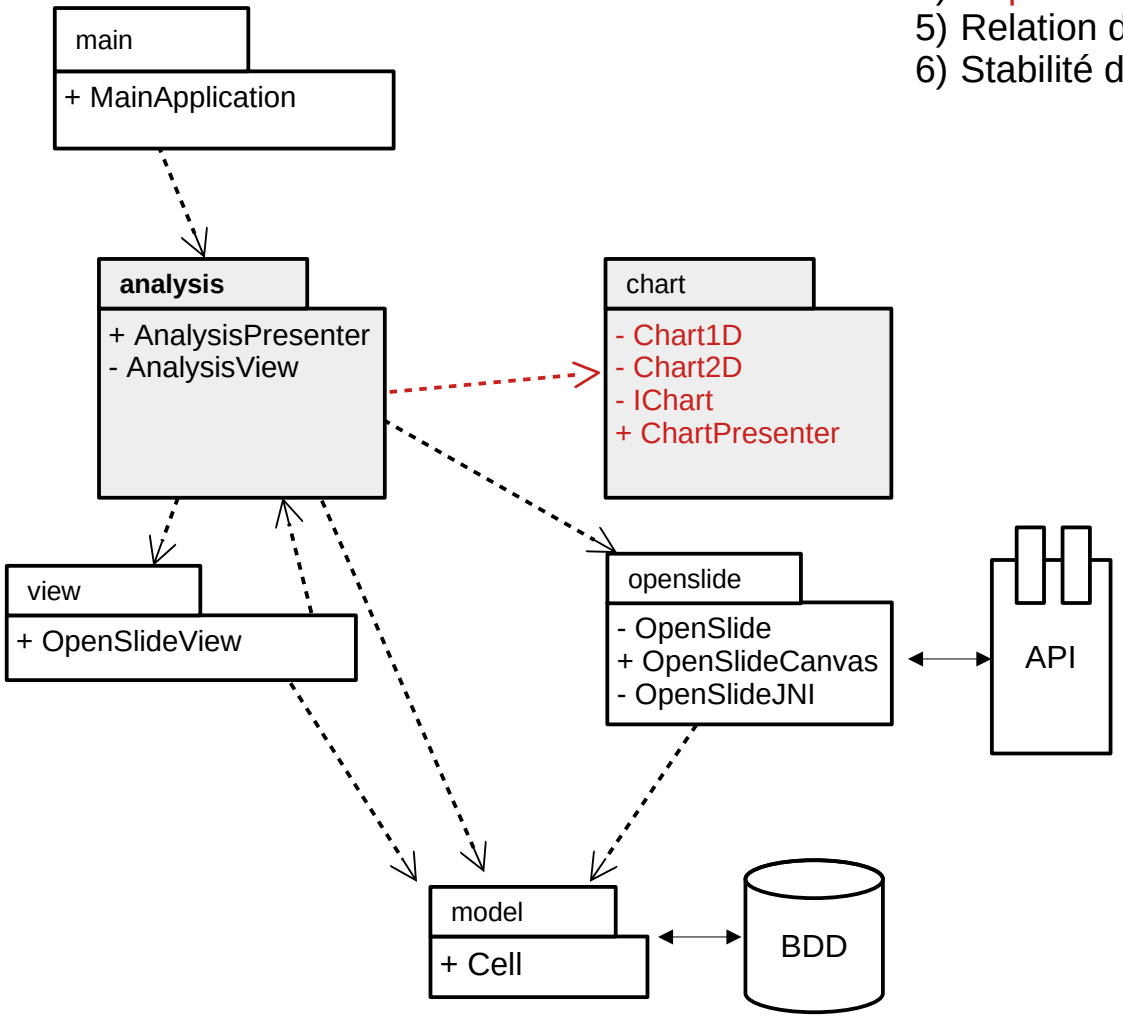
Problème : dépendance acyclique entre presenter et chart

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



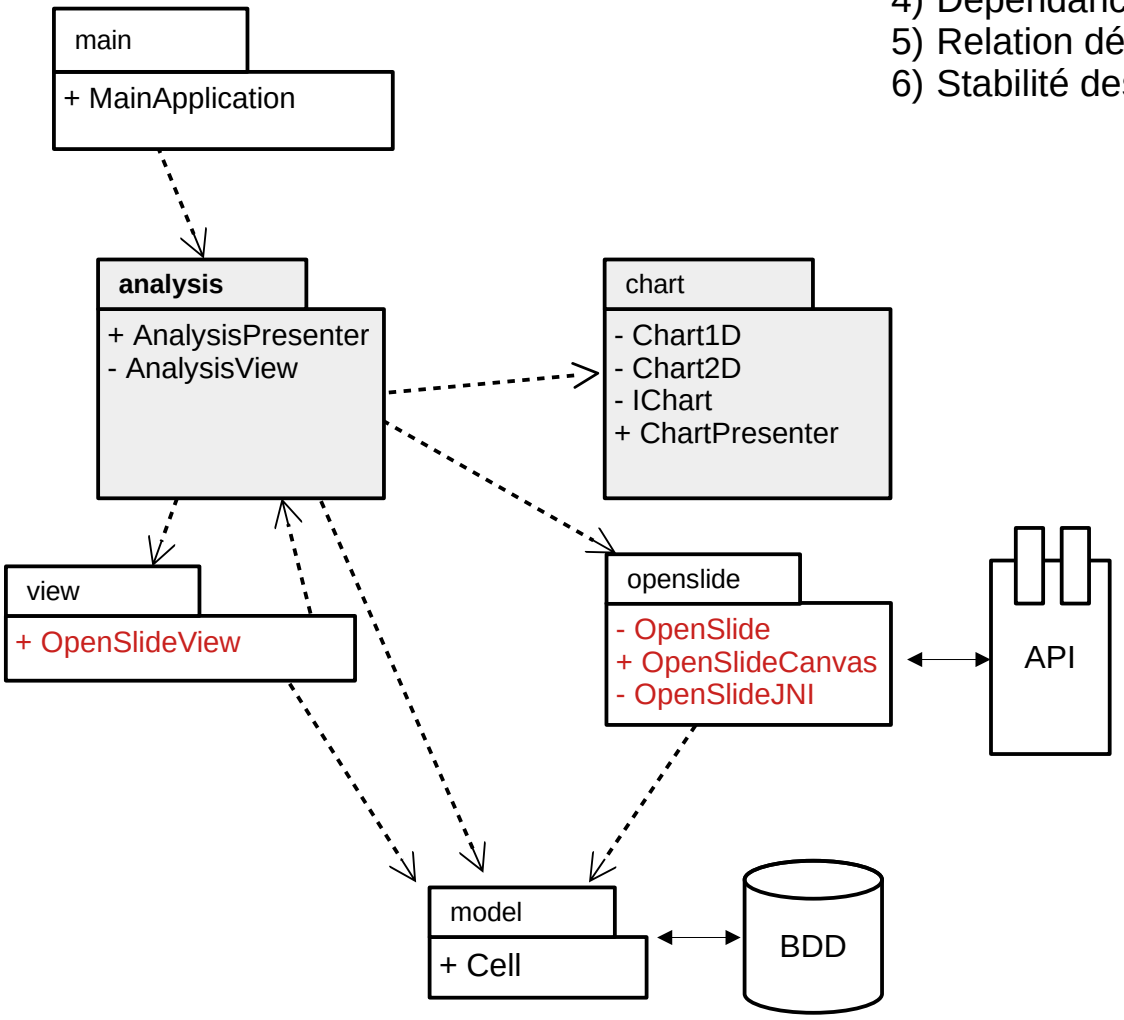
Solution : Créer une classe ChartPresenter à partir du code de AnalysisPresenter.
Remarque : on change la visibilité des classes dans le paquet. Ne reste qu'une seule classe publique. Idem pour le paquet Analysis

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



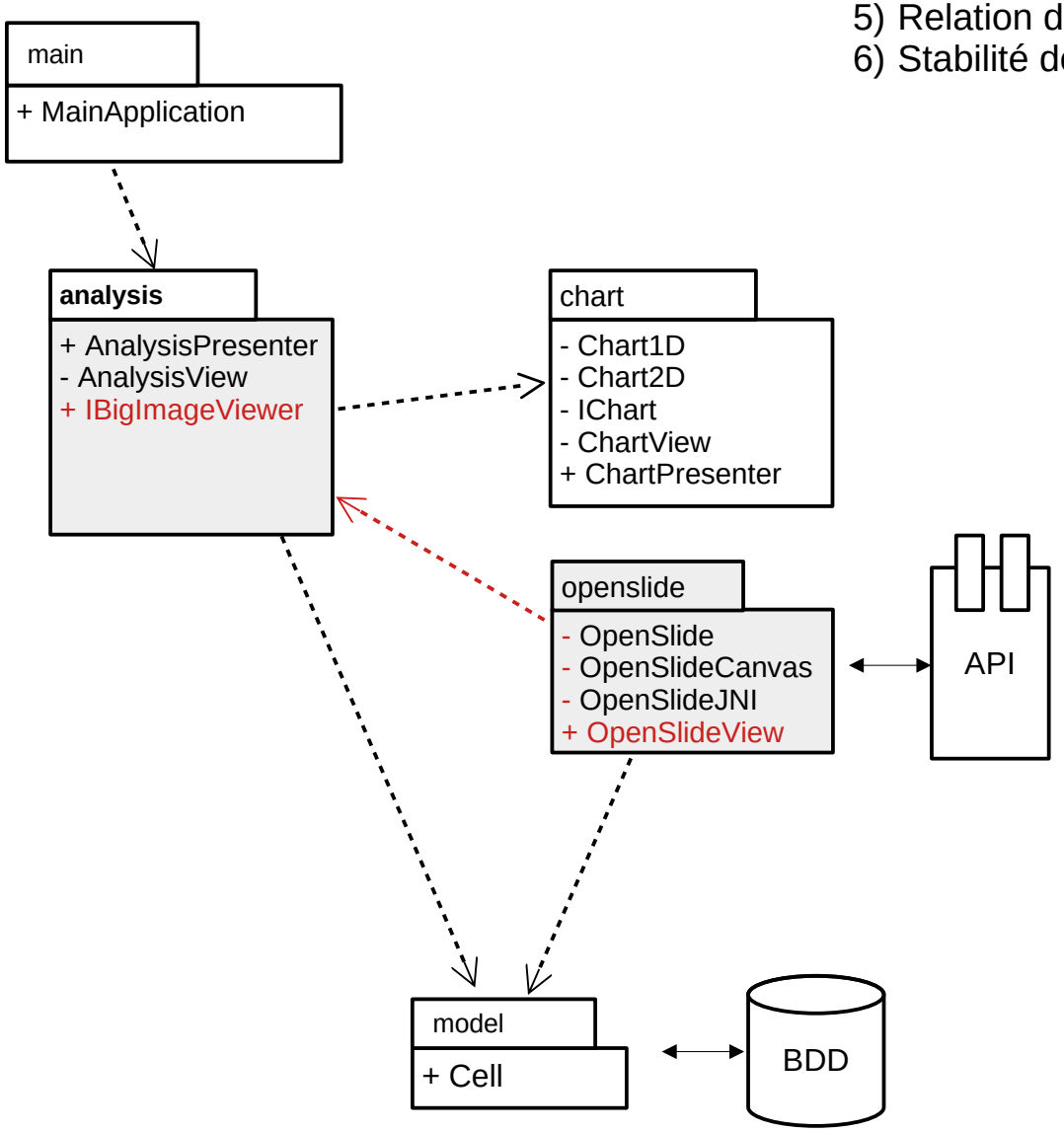
Problèmes : OpenSlide est éclaté dans plusieurs paquets.
OpenSlide est une bibliothèque externe qui est amenée à évoluer. Il doit être possible de changer de bibliothèque.

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



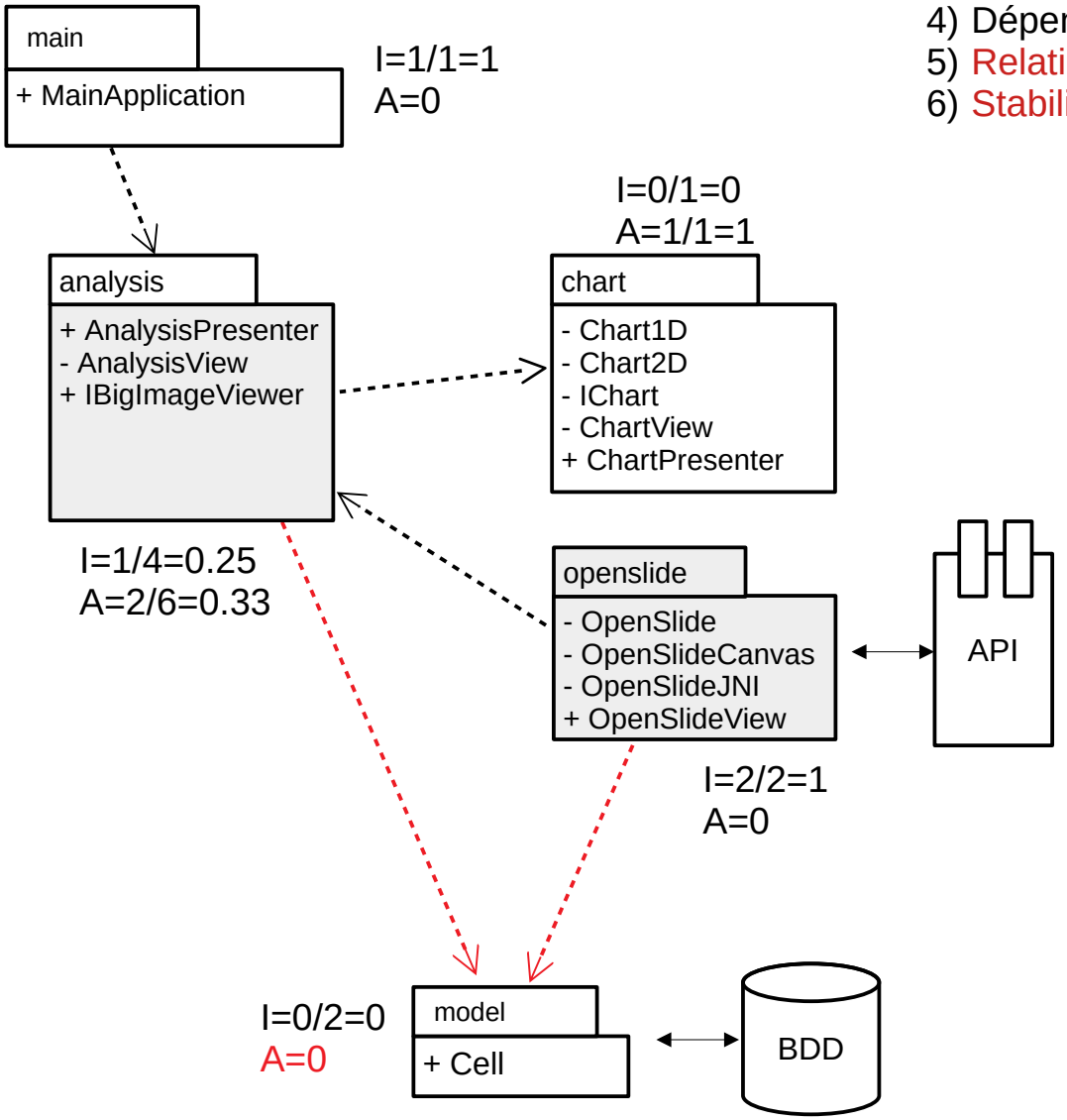
Solution : équivalence réutilisation / livraison pour OpenSlide. Ajouter une interface IBigImageViewer dans la présentateur de l'analyse pour interfacer la bibliothèque.
Changer la visibilité des classes dans le paquet.

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



Problème : relation dépendance / stabilité + stabilité des abstractions

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions



Solution : ajout d'une interface AbstractCell dans le modèle.

Principes
1) Équivalence réutilisation / livraison
2) Fermeture commune
3) Réutilisation commune
4) Dépendances acycliques
5) Relation dépendance / stabilité
6) Stabilité des abstractions