



SURICAT
DATABASE TRANSACTION
VISUALIZER

Denécheau Angélique
Le Maitre Théowen
Jaouen Romain
Guerin Ludwig

Sommaire

- Pourquoi ce projet ?
- Analyse
- Solutions techniques
- Empreinte visuelle
- Progression
- Conclusion

Pourquoi ce projet ?



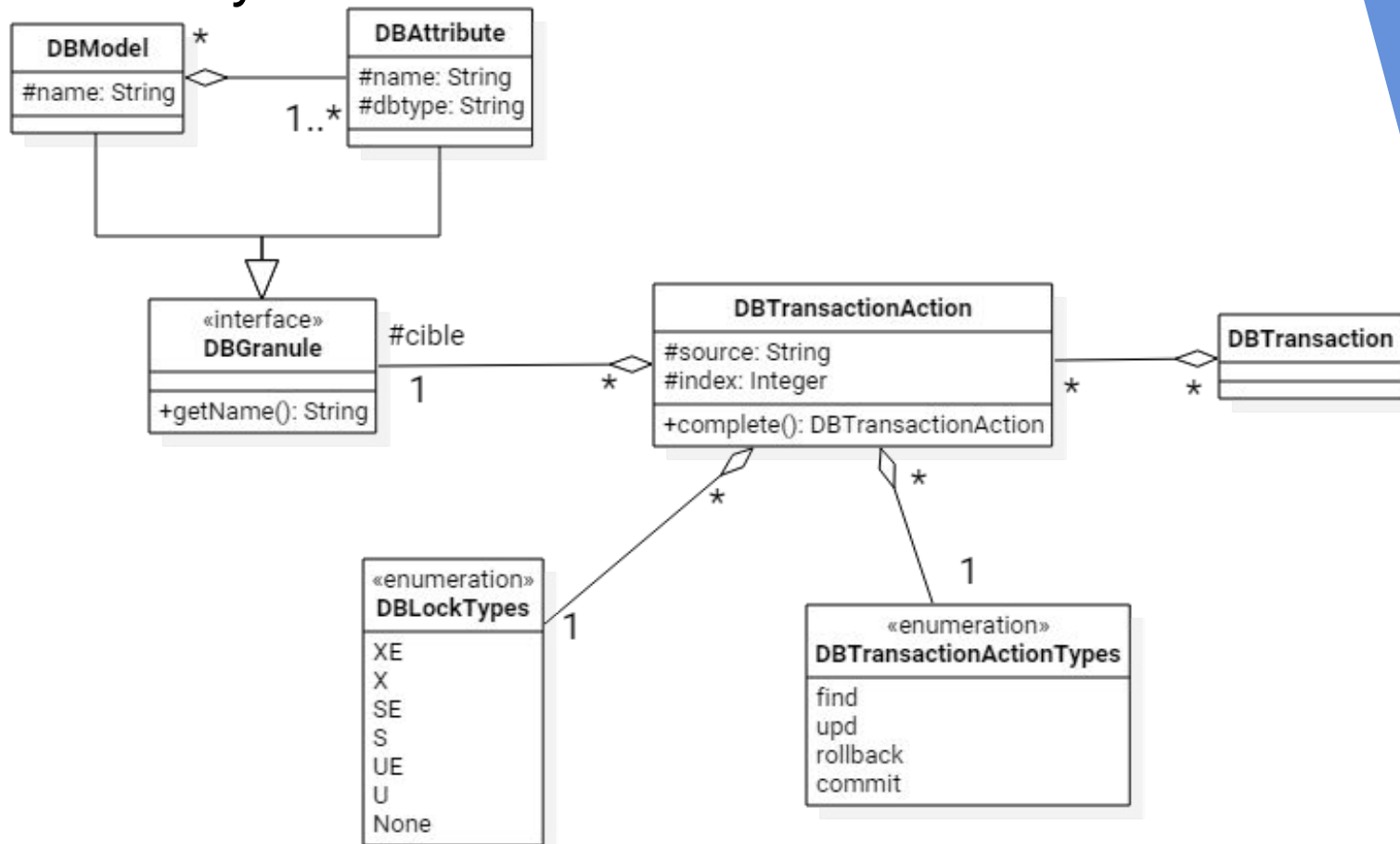
- Base de donnée
- Transactions
- Concurrences
- Analyse



Analyse

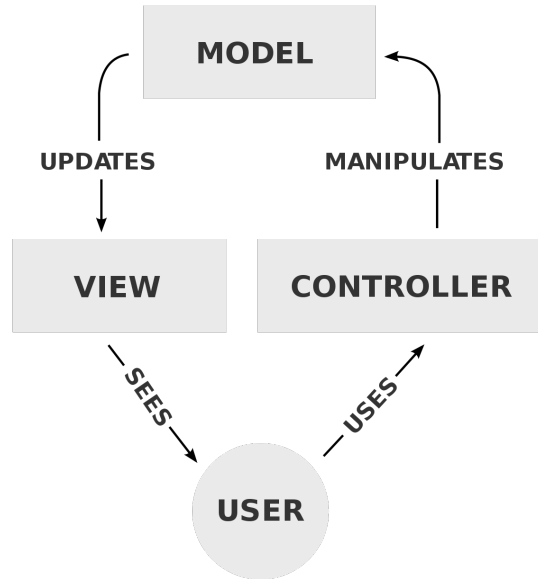


SURICAT



Solutions techniques

- JSON et GSON, pourquoi ?
- Représentation des données, comment ?
- Modèle - Vue - Contrôleur

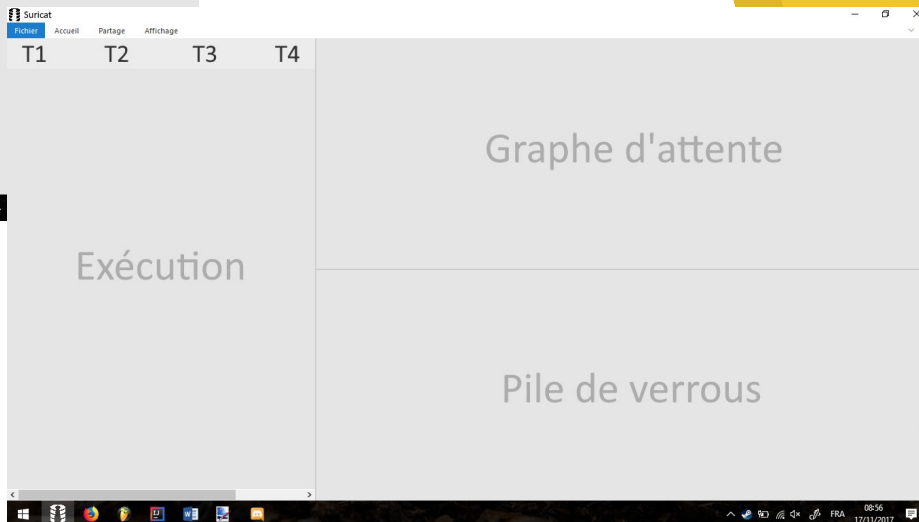


```
"models": [  
  {  
    "name": "R",  
    "attributes": [  
      {  
        "name": "R.A",  
        "dbtype": "VARCHAR"  
      },  
      {  
        "name": "R.B",  
        "dbtype": "SMALLINT"  
      }  
    ]  
  },  
  ]  
},  
  
"transactions": [  
  {  
    "source": "t1",  
    "index": 1,  
    "lock": "XE",  
    "type": "find",  
    "target": "R"  
  },  
  {  
    "source": "t2",  
    "index": 2,  
    "lock": "SE"
```

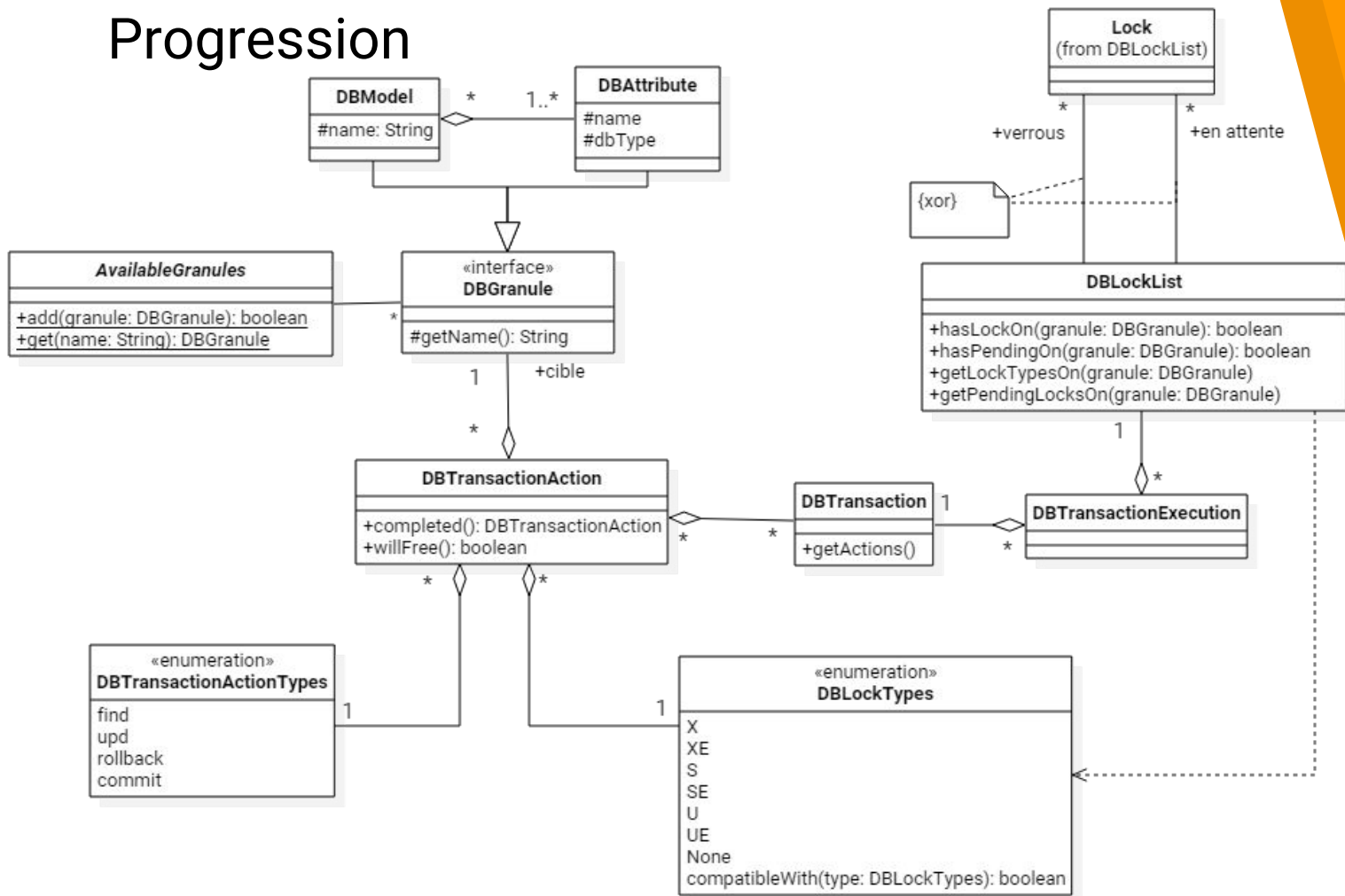
Empreinte visuelle



- Simple
- Minimaliste
- Épuré



Progression



Conclusion



- Besoin de logiciel pour représenter les transactions et leurs concurrences
- Pour faire ce qu'il doit faire, Suricat utilise multiples technologies et paternes de design standardisés tel que le paterne MVC et le format JSON
- Le design global n'a jamais stagné et à constamment évolué durant le projet
- Le projet n'est pas loin d'aboutir à une version réellement finale, la partie principale restante à réaliser est l'expérience utilisateur (la réelle interface visuelle)