

# Git入门简介

分布式VCS的使用及相关规范

# Git是...

- VCS（版本控制系统）或 SCM
- 分布式VCS -- DVCS
- 开源、免费的
- 轻量、快速以及便捷的
- linux之父作品

# Why Git ?!

# 谈谈VSS/CVS/SVN

# 集中式代码库

- 无服务器不提交
- 提交规范难以落实
- 每日构建难

# 抢占式

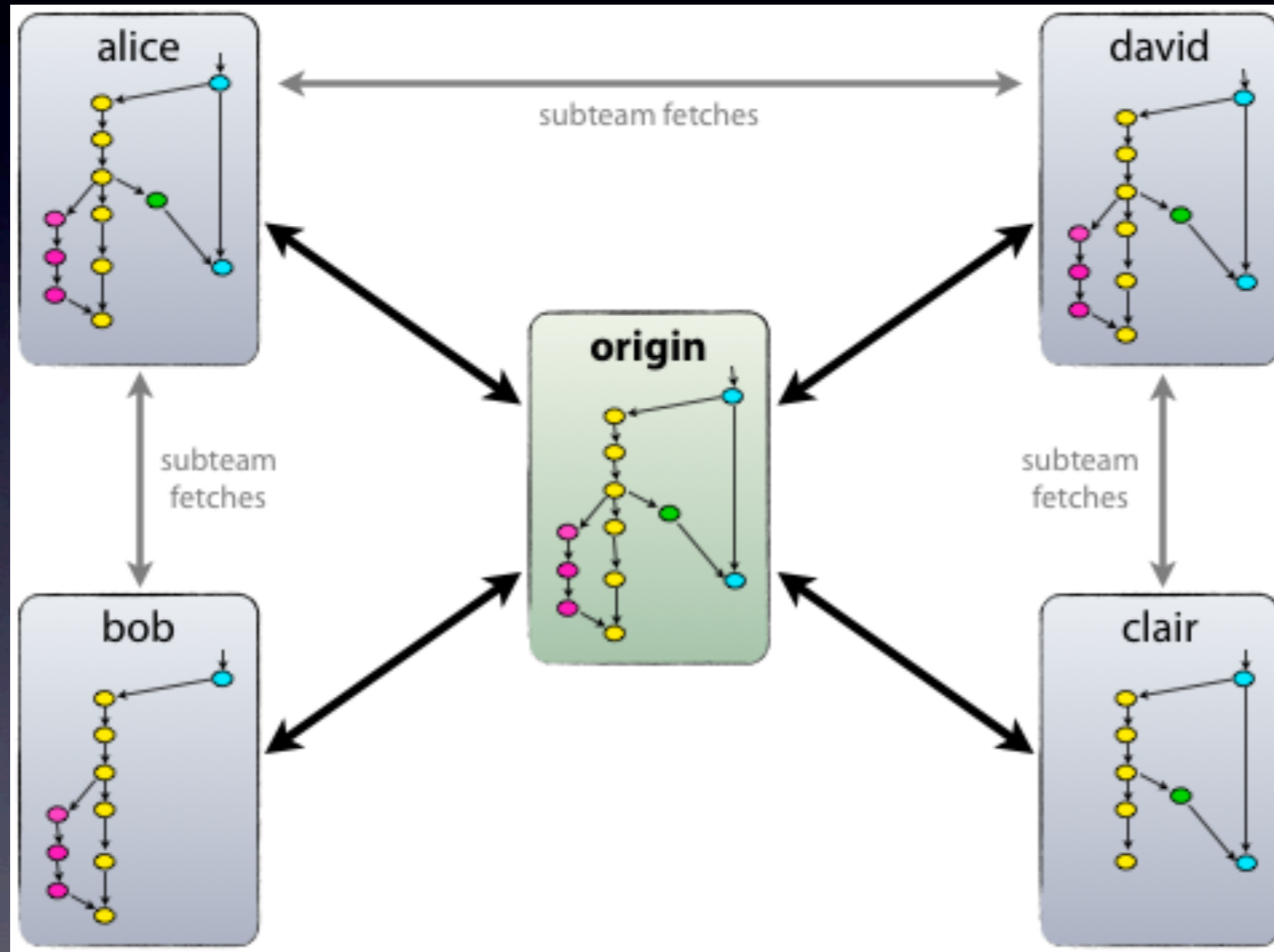
- VSS的默认模式
- 独占 == 保守 == 安全？
- 独占 == 低效！

# 工程文件被污染

- 无处不在的.svn .cvs .vss文件污染

# 谈谈Git/Hg/....

# 分布式代码仓库



# 分布式的好处

- 不依赖服务器，本地实现版本控制
- 0 ... \*n 个服务器，人人都是服务器

# 轻量级的分支

- 分支更容易
- 使用分支更好隔离代码

# 干干净净的目录

工具神马的都是浮云

别当VCS是备份工具

# Git就这么几招

- 本地常用命令
- 远程操作常用命令

# 本地常用命令

- 创建代码仓库
- 添加文件
- 查看仓库状态
- 提交代码
- 忽略某些文件

# 创建代码仓库

- `cd /your-working-dir`
- `git init`

# 添加文件

- `git add main.java`
- `git add .`
- `git add *.js`

# 查看仓库状态

- `git status`
  - 仓库当前工作于哪个分支
  - 有哪些文件已被修改
  - 有哪些文件未被添加至版本控制

# 提交代码

- `git commit main.java`
- `git commit -a`
- `git commit *.js`

# 忽略某些文件

- 编辑 .gitignore 文件：
  - .DS\_Store
  - \*.class
  - \*.jar
  - build
  - dist

# 远程操作命令

- 从远程复制仓库
- 从远程同步仓库
- 提交修改至远程

# 从远程复制仓库

- `git clone git@host:repot.git`
- `git remote add origin git@host:repot.git`
  - `origin` -- 远程仓库别名
  - `git@....` -- 远程仓库

# 从远程同步仓库

- `git pull origin master`
  - `origin` -- 远程仓库别名
  - `master` -- 远程分支名

# 提交修改至远程

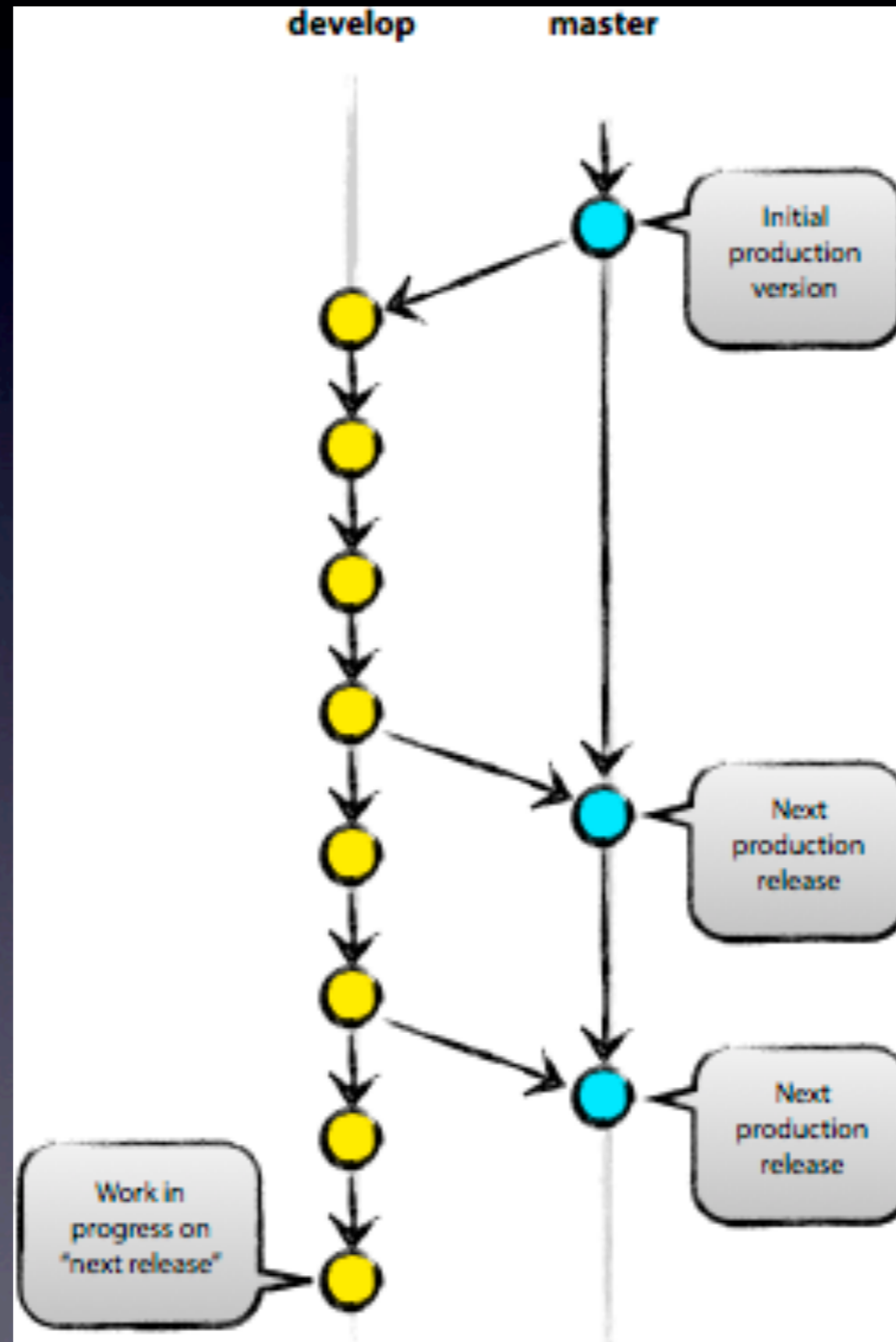
- `git push origin master`
  - `origin` -- 远程仓库别名
  - `master` -- 远程分支名

Show Time × 3

# 版本控制规范

- 双分支结构
- 代码提交守则

# 双分支结构



# develop分支

- 本地develop:
  - 日常工作的主要分支
- 远程develop:
  - 用于daily build
- 新版本发布发合并到master分支

# master分支

- 永远只提供成品级的代码
- 通过tag来管理代码的版本
- 只有新版本发布时，Master才会被用到

# 代码提交守则

- 粒度控制
  - 每次提交都具有完整的意义
  - 无直接关联的文件分批提交
  - 尽量别用 `git commit -a`
- 认真对待提交备注 -- 看的人可能是你
- 编译/运行/测试不通过的代码不能提交至远程

# 善用git-flow

- 让git flow接管分支
  - git flow init
  - 摇滚吧

# 善用git-flow

- 以特性为单位的开发
  - git flow feature start login
    - 添加N个文件
    - 提交N次
  - git flow feature finish login

# 善用git-flow

- 轻松发布新版本
  - git flow release start v1.0
    - commit N 个补丁
    - 修改N个Bug
  - git flow release finish v1.0

# 参考

- <http://git-scm.com/>
- <http://code.google.com/p/msysgit/>
- <http://progit.org/book/zh/>
- <https://github.com/nvie/gitflow/>
- <http://www.gitboxapp.com/>
- <http://gitx.frim.nl>
- <http://www.jeffkit.info/2010/12/860/>
- <http://www.juvenxu.com/2010/11/28/a-successful-git-branching-model/>