Spickzettel: GitHub – Technische Komponenten & Repository-Struktur

1. Wie speichert GitHub Repositories?

- GitHub-Repositories sind **normale Git-Repositories** mit Remote-Hosting.
- Alle Dateien, Commits & Branches sind **dezentral** wie in lokalen Git-Repos.
- Repository-Daten liegen auf verteilten **Git-Servern** (High Availability & Redundanz).
- **Objektspeicherung:** GitHub nutzt optimierte **Pack-Dateien**, ähnlich wie `git gc`.

2. Unterschiede zwischen lokalen und GitHub-Repositories

Feature	Lokal	GitHub
Speicherung	`.git` Ordner auf Festplatte	Verteilt auf Servern
Zugriff	Direkt auf Dateien zugreifbar	Über API oder `git clone`
Authentifizierung	Keine nötig	SSH, HTTPS, PAT erforderlich
Branches	Lokal verwaltbar	Remote-Branches synchronisierbar
Backup	Selbst verwalten	Automatische Replikation & Sicherung

3. Wo liegen meine Repositories auf GitHub?

- Standardmäßig unter: `https://github.com/<nutzer>/<repo>.git`
- Zugriffsmöglichkeiten:
  - **HTTPS:** Einfach, aber erfordert Passwort/Token.
  - **SSH:** Sicher & bequemer für regelmäßige Nutzung.

Repository-URL herausfinden

```
git remote -v
```

- Zeigt alle Remote-Verknüpfungen eines lokalen Repositories.

4. Repository von GitHub klonen

```
git clone <url>
```

- Erstellt eine lokale Kopie eines GitHub-Repositories.

5. Änderungen mit GitHub synchronisieren

- Änderungen hochladen (`push`):

```
git push origin main
```

- Änderungen abrufen (`pull`):

```
git pull origin main
```

- Änderungen ohne automatischen Merge abrufen (`fetch`):

```
git fetch origin
```

6. GitHub API & Automatisierung

- GitHub stellt eine **REST API & GraphQL API** für automatisierte Workflows bereit.
- Beispiele für API-Nutzung:
  - Repositories abfragen
  - Commits & Pull Requests verwalten
  - CI/CD-Workflows automatisieren (z. B. mit GitHub Actions)

Best Practices

- **SSH nutzen**, wenn regelmäßig mit GitHub gearbeitet wird.
- **Remote-URLs überprüfen** (`git remote -v`), um versehentliche Pushes zu vermeiden.
- **Repository regelmäßig aufräumen** (`git gc`, `git repack`) für Performance-Optimierung.
- **GitHub API für Automatisierungen** (z. B. PRs, Status Checks) in Workflows einbinden.