Template as a Code (TAAC)

PART 1 — CLEAR-CUT TECHNICAL IMPLEMENTATION PLAN

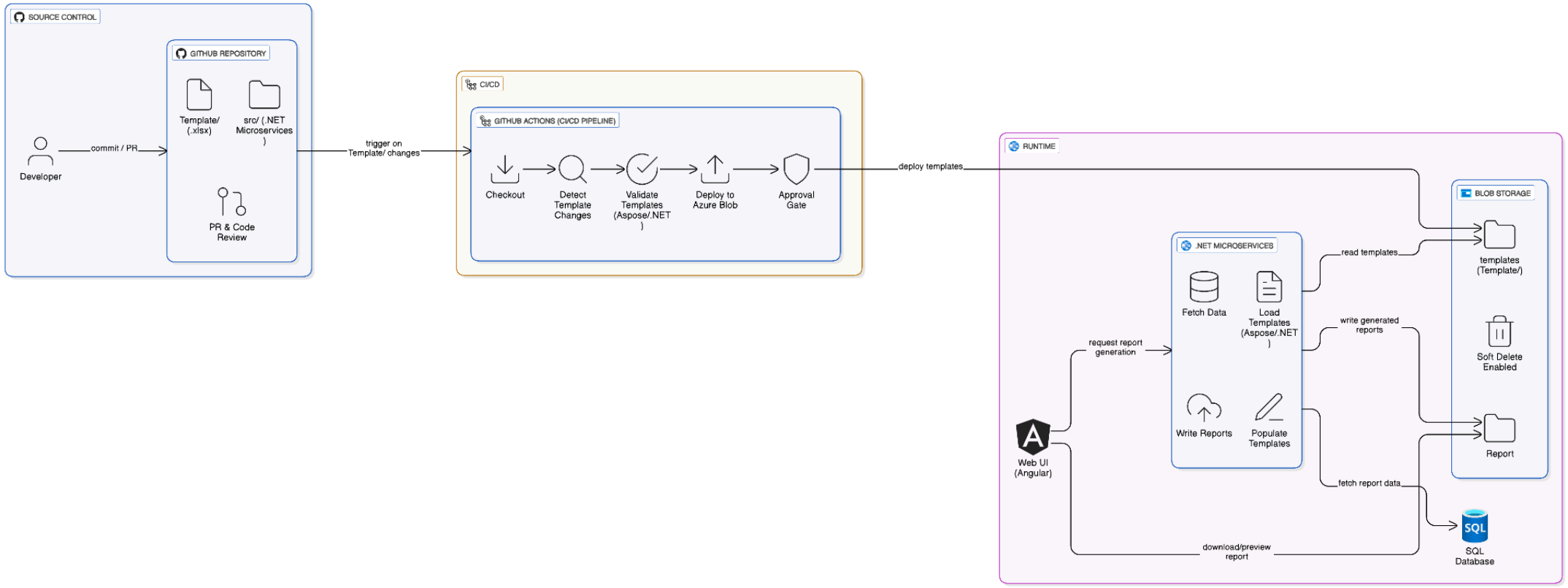
1. Goal, Objective & Assumptions:

- **GitHub is the source of truth for Excel templates.**
- **Azure Blob Storage is a deploy target, not an editing location.**

Objective

Eliminate manual Excel template updates by managing templates via GitHub and deploying them automatically to Azure Blob Storage using CI/CD.

Architecture:



2. Roles & Responsibilities

Developers (Template Owners)

Responsible for **WHAT changes**.

DevOps / Platform Team

Responsible for **HOW changes are deployed safely**.

Flow

1. Developer modifies Excel template in GitHub (`template_migration/`) (already exists)
2. PR reviewed and approved
3. GitHub Actions pipeline triggered
4. Automated template validation
5. Azure Blob Storage synchronized (add/update/delete)
6. .NET microservices load templates at runtime
7. Generated reports stored back in Blob Storage

3. Developer Responsibilities

3.1 Repository Structure (Developer-Owned)

Developers **must only work inside Git**.

```
repo-root/
├── src/
│   └── .NET microservices
│   ├── template_migration/
│       ├── TaxReport.xlsx
│       ├── VehicleSummary.xlsx
│       ├── PayrollFile.xlsx
│       └── README.md
```

Rules

- No subfolders
- No version folders
- One file = one template
- File name is the template identifier

3.2 How a Developer Changes a Template

Step 1 — Pull Latest Code

```
git pull origin develop
```

Step 2 — Modify Template Locally

- Open Excel template locally
- Make formatting / formula / layout changes

3.3 Mandatory Local Validation

Developer must ensure:

- File opens without corruption
- All required sheets exist
- Named ranges still exist in Constants and other config sheets
- No broken formulas

3.4 Commit & PR Discipline

```
git add template_migration/TaxReport.xlsx
git commit -m "Fix rounding issue in TaxReport template"
git push origin feature/tax-report-fix
```

PR requirements

- Description of what changed
- Impact assessment (data/layout)

4. DevOps Responsibilities

4.1 CI/CD Ownership

- GitHub Actions workflows
- Azure authentication
- Blob lifecycle policies
- Deployment approvals

4.2 GitHub Actions Trigger Strategy

Pipeline runs **only** when templates change:

```
on:
  push:
    paths:
      - "template_migration/**"
```

4.3 Pipeline Stage Breakdown

Stage 1 — Checkout & Change Detection

Detect exactly what changed:

git diff --name-status HEAD~1 HEAD -- Template/

Outputs:

- `A` → new template
- `M` → updated template
- `D` → deleted template

Stage 2 — Automated Template Validation

Validation shall runs **before any upload** but there will be no validations as of now to validate template(s).

Stage 3 — Azure Blob Sync

Add / Update (Overwrite Allowed)

```
az storage blob upload \
  --container-name templates \
  --source template_migration/TaxReport.xlsx \
  --name Template/TaxReport.xlsx \
  --overwrite true
```

Delete (Controlled)

```
az storage blob delete \
  --container-name templates \
  --name Template/OldReport.xlsx
```

4.4 Delete Protection

DevOps **must enable**:

- **Blob Soft Delete** (minimum 30 days)
- Optional: manual approval gate if delete detected

This guarantees recovery even if Git deletion was accidental.

4.5 Environment Protection

- Separate storage accounts or containers for:
    - Dev
    - Stage
    - PPROD
    - Prod
- Production deploys require **approval**
- CI/CD identity uses **Managed Identity / Service Principal**

5. Runtime Behavior (No Changes Needed to make it run currently, but…)

Our existing logic continues to work:

Template/{TemplateName}.xlsx

- Always reads latest template
- No version awareness needed
- No redeploy required for template change

Required Code change to read Azure Storage container folders “Template” and “Report” from appsettings rather than AppConstants.cs:

- Add “template” and “Report” in appsettings of services where “AppConstants” C# file has these entries.
- Delete these folder names from AppConstants
- Unit test to make sure container are referred correctly and templates are being read and reports are able to be written and read.
- This will give independency to Devops team in future to control container folder names without any code change required at backend side.

6. Rollback Model

Rollback = **Git revert + pipeline run**

Steps:

1. Revert commit in GitHub
2. Merge PR
3. Pipeline re-syncs Azure Blob
4. Previous template restored

Typical rollback: **< 30 minutes for develop**

7. Governance & Audit

Control	Coverage	Responsible
Who changed	Git commit	Developer
Why	PR description	Developer
When	Git history	Developer
What deployed	CI/CD logs	Devops
Recovery	Blob soft delete	Devops

This fully satisfies audit and compliance reviews.