# The agent builds your report. It never sees your numbers.

AI agents now build reports in minutes. To do that, most tools hand them the data as well: revenue, margin, costs, customer lists. Every value ends up with the model provider. For source code that is often acceptable. For **company figures** it usually is not. Bino keeps the two apart: the agent writes the **blueprint** of the report. The real numbers join only in the **build**, without AI.

## 01 Code is not data

| DATA IN THE PROMPT | ONLY CODE IN THE PROMPT |
|---|---|
| To build a chart, the agent gets the spreadsheet or the database login. Every number goes to the model. | The agent writes YAML and SQL. It knows the columns and the structure, but not the actual values. |
| **REPORT = RESULT** | **REPORT = BLUEPRINT** |
| The numbers are on the generated slide. Design and data are now inseparable. | The manifests only describe the report. Design and data meet in `bino build`, not before. |
| **REAL DATA FOR TESTING** | **SYNTHETIC DATA FOR TESTING** |
| The agent needs data to see the layout. In practice, that means the real data. | A representative dataset with the same columns. The layout is finished with it. |

**WHAT THE AGENT SEES**

- **YAML manifests** · pages, tables, charts
- **SQL queries** · which column goes where
- **Schema** · column names and types
- **Synthetic data** · values with no real meaning
- **Preview and layout checks** · with those values

**WHAT THE AGENT NEVER SEES**

- **Real figures** · not one row, not one total
- **Credentials** · passwords live only in CI
- **The final PDF** · it goes straight to the readers

## 02 Five rules for agentic reporting without data exposure

None of them needs a better model or a contract with the provider. They are properties of the format and the process.

### I Separate

Design and data are **separate files**. They meet only in the build.

### II Representative

Synthetic data has the **same columns**, magnitudes and edge cases as the real data.

### III Swappable

Both sources have the **same name**. SQL and layout do not change when you switch.

### IV Built without AI

The final PDF is built in **CI**, with no model in the process.

### V Isolated

Real data and **credentials** exist only where the build runs.

**THE REAL QUESTION**

Not "may the AI see our data?" but **does it need to at all?** To design a report, you only need to know the structure of the data. Only the build needs the values, and the build does not need AI.

## 03 One data source, two faces

In Bino a report is **plain text**: YAML for pages and charts, SQL for the data. A `DataSource` only says **where** the data comes from. So there can be two sources with the same name: a synthetic one and a real one. A **constraint** decides at build time which one applies.

Without any setting, Bino uses the synthetic source. That holds for `bino preview` and for every `bino build` on the laptop, also when the agent starts it through `bino mcp`. Only CI sets `REPORT_DATA=real` and has the database password.

SQL queries and layout are the same in both cases. What the agent finished with synthetic data, CI renders with real data **without a single change**.

**LOCALLY AND IN CI**

```
# locally, with the agent: synthetic
bino preview

# in CI, without AI: real data
env:
  REPORT_DATA: real
  DB_PASSWORD: ${{ secrets.DB_PASSWORD }}
run: bino build --data-validation=fail
```

**DATASOURCES/SALES.YAML · EXCERPT**

```
apiVersion: bino.bi/v1alpha1
kind: DataSource
metadata:
  name: sales
  constraints: [labels.data==synthetic]
spec:
  type: csv
  path: ./data/synthetic/sales.csv
---
apiVersion: bino.bi/v1alpha1
kind: DataSource
metadata:
  name: sales  # same name
  constraints: [labels.data==real]
spec:
  type: postgres_query
  connection:
    host: ${DB_HOST:localhost}
    port: 5432
    database: finance
    user: reporting
    secret: dbCredentials # reads DB_PASSWORD
  query: SELECT * FROM fact_sales
---
apiVersion: bino.bi/v1alpha1
kind: ReportArtefact
metadata:
  name: monthly
  labels:
    data: ${REPORT_DATA:synthetic}
```

| 1 · local | 2 · local | 3 · gate | 4 · gate | 5 · CI | 6 · gate | 7 · output |
|---|---|---|---|---|---|---|
| **Agent writes YAML + SQL** `bino mcp` | **Preview with synthetic data** `bino preview` | **Check the layout** `get_layout_state` | **A person reads the diff** pull request in Git | **Build with real data** `REPORT_DATA=real` | **Checks without AI** `--data-validation=fail` | **PDF to the readers** never back to the agent |

`if CI fails, the error description goes back to step 1, never the data rows ↺`

### WHAT MATTERS

- **Include edge cases.** Negative values, gaps, long names, and large numbers. If the synthetic data lacks them, the layout breaks only with the real data.
- **Structure is visible.** The agent knows column names and types. If even the names are confidential, use neutral ones.
- **Only synthetic sources locally.** `get_rows` and `introspect_source` show data rows to the agent. On its machine they must find only synthetic data.
- **Open no way back.** The PDF, build logs and CI output do not go to the agent.

### THE ACTUAL PROTECTION

The constraint in the YAML file is a **switch, not a lock**. The data is safe because credentials and real files exist only in CI.

If the agent sets `REPORT_DATA=real` locally, the build fails: the database is out of reach and the password is missing. The agent gets an error message, but not a single number.

We review a **process**, not every single agent run. This makes it easier for data protection and IT to sign off.