Council Tools — `export_document` Change Log

Date: 2026-11-03 · **Host:** ai1 · **Status:** Live & verified

Goal

Expose existing PDF/DOCX/XLSX export as a council-tools MCP tool, without duplicating logic, returning a download link by default and optionally posting it to the caller's OC channel.

What was added

A single thin MCP tool in `/mnt/nvme/council/oc_mcp.py`:

```
@mcp.tool()
def export_document(content: str, title: str, format: str = "pdf",
                    post_to_channel: bool = False,
                    user_id: str = None, ctx: Context = None) -> str:
    ...
```

Plus constant `DOC_URL = "http://localhost:5051"`. It proxies to the existing `doc-api` service (`POST /generate`) which already builds all three formats from markdown. No changes to `notify_api.py`.

Behavior decisions

• **Default `post_to_channel=False`** — returns the download URL only.
• **User-info routing**: OWUI's council connection sends header `x-openwebui-user-id: {{USER_ID}}`; `_caller_id(ctx)` reads it, resolves the OC webhook via `get_oc_channel`, and posts the link. Same proven path used by `generate_image`.
• **No silent failures**: if posting is requested but the caller/webhook cannot resolve, the result appends `(WARNING: could not post to channel: ...)`. On success it reports `Posted to OC channel.`

Files changed

File	Change
`/mnt/nvme/council/oc_mcp.py`	Added `DOC_URL`, `export_document` tool, non-silent post handling
`/home/owner/install_council.sh`	Heredoc updated (constant, tool, port-substitution sed line, SOUL.md tool inventory)

Verification

• `bash -n install_council.sh` passes; regenerated heredoc compiles; `DOC_API_PORT_PLACEHOLDER` resolves correctly.
• `council-mcp.service` restarted and active after each change.
• `tools/list` over :8765 includes `export_document`.
• Live calls produced real files with working download URLs (PDF and XLSX round-trips).