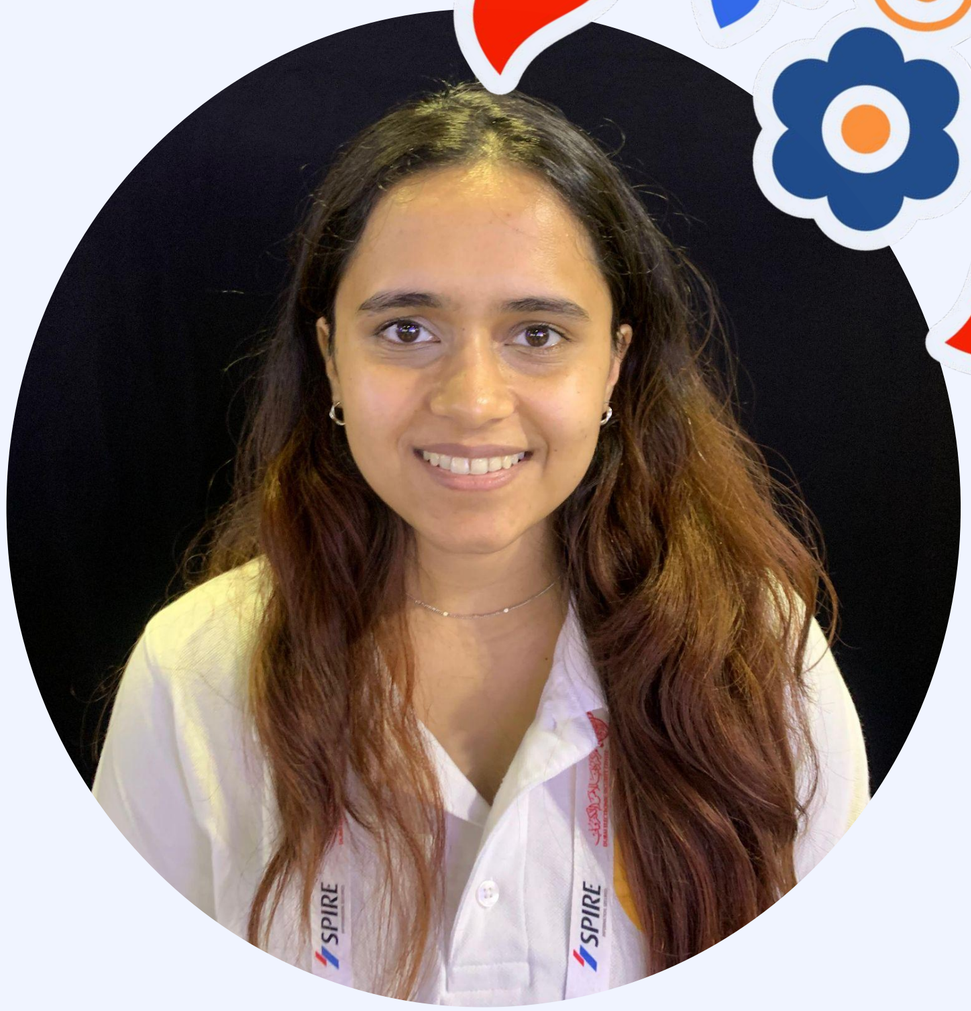
WordCamp Europe
Kraków 2026

AI GOVERNANCE & POLICY ENFORCEMENT FOR ABILITIES API

Anukasha Singh
miniOrange

WordCamp Europe
Kraków 2026

Hi, I am Anukasha!



Sr. Software Engineer, 6 years in cybersecurity and enterprise IAM solutions on WP and other platforms. Curious about AI Identities now!

WHAT ARE THE ABILITIES API?

WordCamp Europe Kraków 2026

Abilities let you **describe a skill or functionality** your site can perform — with defined inputs, outputs, and permissions. (WP 6.9+)

Core abilities shipping in WP 7.0

- `core/get-site-info`
- `core/get-user-info`
- `core/get-environment-info`

WooCommerce — Orders

- `woo/get-order-info`
- `woo/process-refund`
- `woo/update-stock`

WooCommerce — Marketing

- `woo/create-coupon`
- `woo/get-sales-report`
- `woo/update-product`

REGISTERING AN ABILITY

WordCamp Europe Kraków 2026

wp_register_ability()

Each ability needs:

- **label** + **description** — human-readable name
- **category** — e.g. 'marketing'
- **input/output schema** — JSON Schema
- **execute_callback** — the actual logic
- **permission_callback** — who can call it

```
wp_register_ability(
  'woo/create-coupon',
  array(
    'label'            => 'Create Coupon',
    'description'      => ‘Creates a discount coupon',
    'category'         => 'marketing',
    'input_schema'     => array(),
    'output_schema’    => array(),
    'execute_callback'     =>'woo_create_coupon',
    'permission_callback' =>
      function () {
        return current_user_can('manage_woocommerce');
      },
  )
);
```

WordCamp Europe
Kraków 2026

Abilities registered!
What's next?

ABILITIES API - THE AI BRIDGE

WordCamp Europe
Kraków 2026

AI agents speak **natural language**. WordPress speaks **REST APIs**.
Abilities are the **machine-readable bridge.**

With MCP in WP 7.0, Abilities become tools for agents like Claude, ChatGPT, or any agent.

You can prompt Claude to create discount coupons for your store!!

WHAT IS MCP?

CLAUDE — A MARKETING AGENT

WordCamp Europe Kraków 2026

Sarah (Shop Manager) connects **Claude** to her WooCommerce site to create coupons.

Her site has 3 abilities registered with `manage_woocommerce`:

`woo/create-coupon` · `woo/process-refunds` · `woo/update-price`

Sarah prompts: *"Get the store ready for our spring sale."*

Claude ran **all three abilities** — because Sarah's role has access to everything.

PROBLEM?

No scoping — One prompt triggered coupons, refunds, and price changes

No identity — Refunds and price changes show "Sarah" in logs

No audit trail — Did Sarah drop those prices, or the agent?

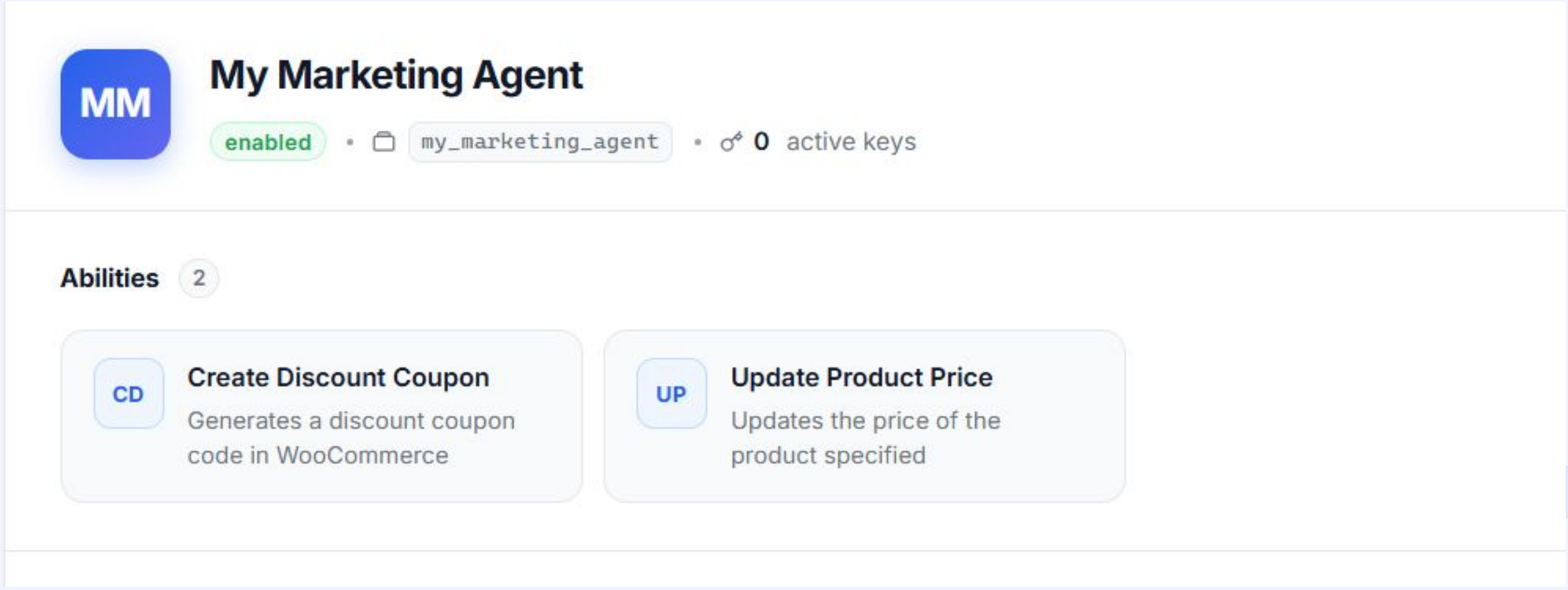
Shared credentials — Leaked app password = full store access

Agents need **smarter permissions** to make them **accountable** and **traceable**.

WordCamp Europe
Kraków 2026

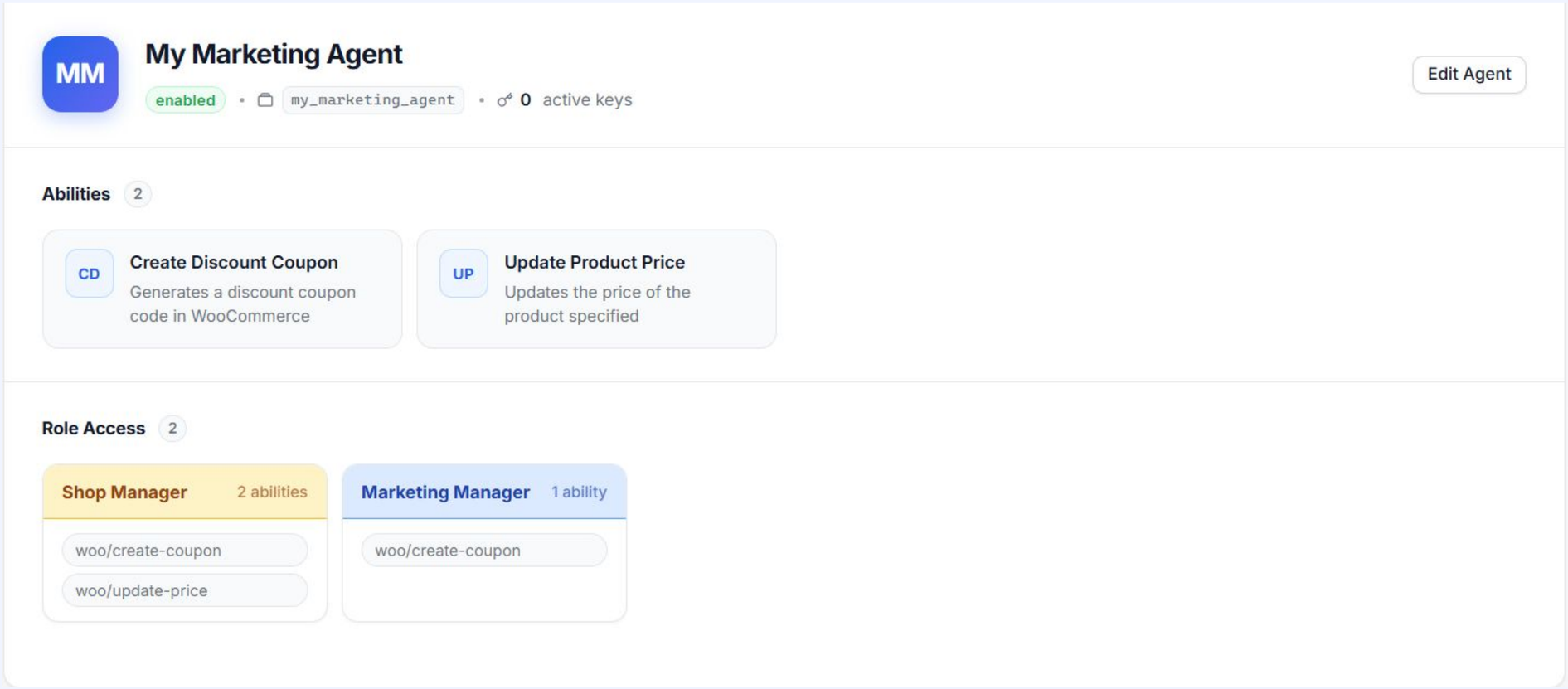
Solution: Running Abilities with Smarter Permissions for Agents

Assign Abilities per Agent



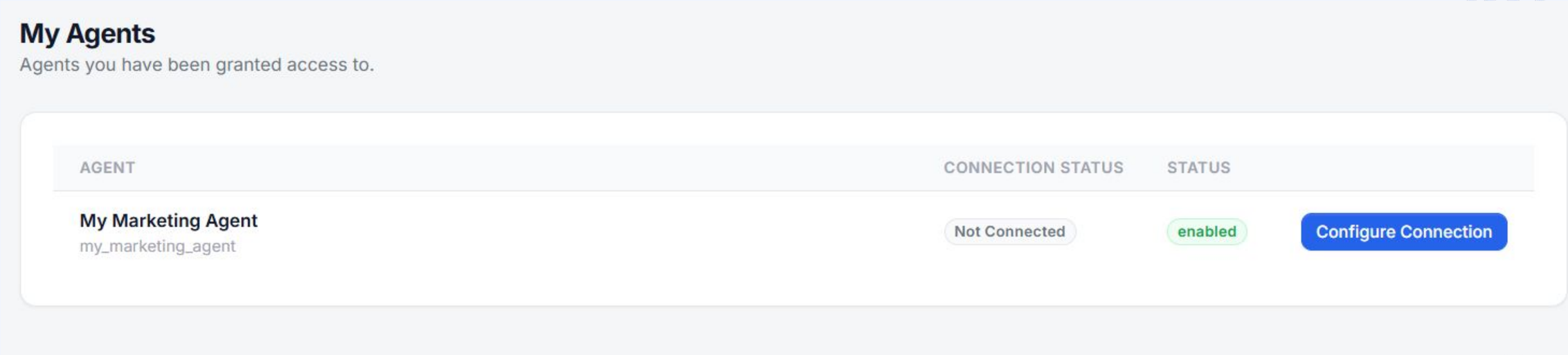
Set Abilities per WP Role

WordCamp Europe
Kraków 2026

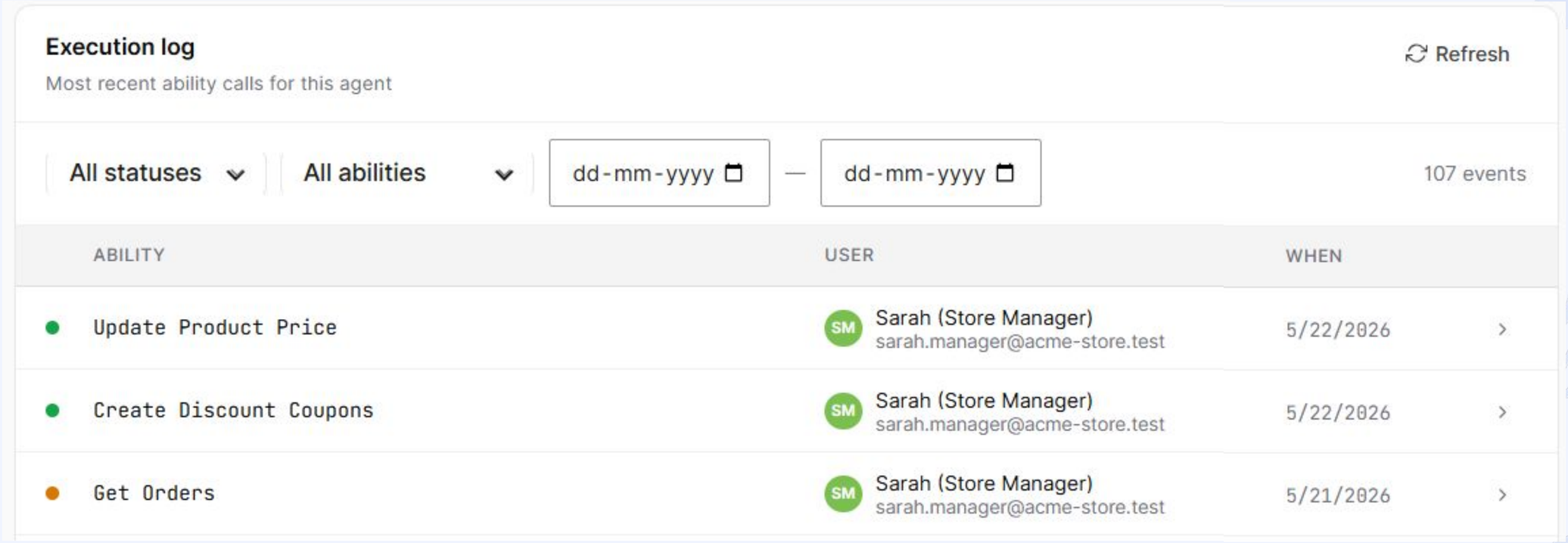


Every user creates their own credentials for the agent

Sarah (Store Manager) logs in, and is asked to configure connection with the agent



Every action is logged



WordCamp Europe
Kraków 2026

THANK YOU!

Connect with me on LinkedIn:

Solution: Running Abilities with Smarter Permissions for Agents

WordCamp Europe
Kraków 2026

1. **Register an agent and assign abilities it can access**
   - `woo/create-coupon` · `woo/update-price`

2. **Set abilities per role**
   Marketing Manager → create-coupon
   Shop Manager → create-coupon + update-price
   Content Creator → no abilities access

3. **Every user creates their own credentials for the agent**

4. **Log every action**
   Agent → user → abilities used → full audit log.

PERMISSIONS, UNTANGLED

WordCamp Europe
Kraków 2026

THE OLD WAY

```
// WooCommerce admin
current_user_can( 'manage_woocommerce' );

// REST endpoint
current_user_can( 'edit_shop_coupons' );

// AJAX handler
current_user_can( 'manage_woocommerce' );
```

⚠️ 3 places, 2 different caps. Which one is right?

THE NEW WAY

```
wp_register_ability(
  'woo/create-coupon',
  ['permission_callback' => function() {
      return current_user_can('manage_woocommerce');
  }]
);

// define once, check anywhere
if ( wp_current_user_can_ability( 'woo/create-coupon’ ) )
{
    // user has permission
}
```

✓ 1 definition, works everywhere