Entity access for lists

A crucially missing piece of the puzzle

Kristiaan Van den Eynde
Kristiaan Van den Eynde
Senior Drupal developer
@Magentix
Kristiaan Van den Eynde
- Work at Factorial GmbH in Hamburg
- Live near Antwerp, Belgium
- Group module maintainer
- Happily married, recently a dad
- Highly sensitive person
Definition of list access
Definition of list access

- Checks access **before** entities are loaded
Definition of list access

- Checks access before entities are loaded
- Takes caching into account
Definition of list access

- Checks access before entities are loaded
- Takes caching into account
- Supported by Views
“But we already have that, so what's the big deal?”

Someone in the audience
Current implementation

The node grants system
What is the node grants system?
What is the node grants system?

- Saves business logic to the database upon node manipulation
What is the node grants system?

- Saves business logic to the database upon node manipulation
- Alters queries tagged with node_access to check for access against this saved business logic
What is the node grants system?

- Saves business logic to the database upon node manipulation
- Alters queries tagged with node_access to check for access against this saved business logic
- Also used as fallback if regular access checks are indecisive
What is the node grants system?

- Saves business logic to the database upon node manipulation
- Alters queries tagged with node_access to check for access against this saved business logic
- Also used as fallback if regular access checks are indecisive

- Bad metaphor: Locks and keys
What is the node grants system?

- Saves business logic to the database upon node manipulation
- Alters queries tagged with node_access to check for access against this saved business logic
- Also used as fallback if regular access checks are indecisive

- Bad metaphor: Locks and keys
- Better metaphor: Bouncers at a night club
What's wrong with node grants?
What's wrong with node grants?

- Only work for "content" (aka nodes)
What's wrong with node grants?

- Only work for "content" (aka nodes)
- Only work for view, update and delete actions
What's wrong with node grants?

- Only work for "content" (aka nodes)
- Only work for view, update and delete actions
- As a result does not scale well for other entities
What's wrong with node grants?

- Only work for "content" (aka nodes)
- Only work for view, update and delete actions
- As a result does not scale well for other entities
- Might try and store extremely complex access logic in the DB
Can it be fixed?
Can it be fixed?

- Not really, the concept is past its due date
- Would either require a new column on existing table or one table per entity type, both are far from ideal
- Would require a new column per supported operation, which again would lead to unwieldy (and buggy) code
Intermezzo

Possible approaches
Keep altering queries

Pros and cons
Keep altering queries

Pros and cons

- Pro: We already have ENTITY_TYPE_access query tags
Keep altering queries

Pros and cons

- Pro: We already have ENTITY_TYPE_access query tags
- Pro: People are already used to this approach
Keep altering queries

Pros and cons

- Pro: We already have ENTITY_TYPE_access query tags
- Pro: People are already used to this approach
- Con: Complicated use cases may find themselves limited by SQL
Scalable pagination

Access checks in code
Scalable pagination

Access checks in code

- Proposed by catch early 2017
- Based on a Four Kitchens blog post from 2009
Scalable pagination explained
Scalable pagination explained

- You ask for more results than you need and pull them through your access logic
Scalable pagination explained

- You ask for more results than you need and pull them through your access logic.
- If you do not have enough results, go back to the database for more.
Scalable pagination explained

- You ask for more results than you need and pull them through your access logic
- If you do not have enough results, go back to the database for more
- Keep track of the first and last item and use them for paging
Scalable pagination explained

- You ask for more results than you need and pull them through your access logic
- If you do not have enough results, go back to the database for more
- Keep track of the first and last item and use them for paging
- Works best on sites where most content is accessible to everyone
Scalable pagination

Pros and cons
Scalable pagination

Pros and cons

- **Pro:** Same access logic for both individual entities and entity lists
Scalable pagination

Pros and cons

- Pro: Same access logic for both individual entities and entity lists
- Pro: No "content drift" due to Reddit-style pagers (next/previous)
Scalable pagination

Pros and cons

- Pro: Same access logic for both individual entities and entity lists
- Pro: No "content drift" due to Reddit-style pagers (next/previous)
- Con: Poor performance on sites with more complex access set-ups
Scalable pagination

Pros and cons

- Pro: Same access logic for both individual entities and entity lists
- Pro: No "content drift" due to Reddit-style pagers (next/previous)
- Con: Poor performance on sites with more complex access set-ups
- Con: No indication of amount of possible results
Scalable pagination

Pros and cons

- **Pro:** Same access logic for both individual entities and entity lists
- **Pro:** No "content drift" due to Reddit-style pagers (next/previous)
- **Con:** Poor performance on sites with more complex access set-ups
- **Con:** No indication of amount of possible results
- **Con:** People are not familiar with this approach
"You don't seem to be a fan of scalable pagination"

Someone else in the audience
Query altering: Part Deux
A summary of previous work
A summary of previous work

- Extend the entity access system with a new grants API (and deprecate the query-alter-based node grants API)
  https://www.drupal.org/project/drupal/issues/777578
A summary of previous work

- Extend the entity access system with a new grants API (and deprecate the query-alter-based node grants API)
  https://www.drupal.org/project/drupal/issues/777578
- Entity access policies
  https://www.drupal.org/project/entity_access_policies
A summary of previous work

- Extend the entity access system with a new grants API (and deprecate the query-alter-based node grants API)
  https://www.drupal.org/project/drupal/issues/777578
- Entity access policies
  https://www.drupal.org/project/entity_access_policies
- Implement a query-level entity access API
  https://www.drupal.org/project/entity/issues/2909970
Entity access policies
Entity access policies

- A collection of access plugin, e.g.: is_published
Entity access policies

- A collection of access plugin, e.g.: is_published
- Used in policy config entities that list which entity types and operations they apply to
Entity access policies

- A collection of access plugin, e.g.: is_published
- Used in policy config entities that list which entity types and operations they apply to
- When an entity query is launched, this system kicks in, finds all applicable policies and compiles them into one query alter
Entity access policies

- A collection of access plugin, e.g.: is_published
- Used in policy config entities that list which entity types and operations they apply to
- When an entity query is launched, this system kicks in, finds all applicable policies and compiles them into one query alter
- You can build a UI showing all of the active access policies for your website and even allowing you to edit them
Entity access policies

Pros and cons
Entity access policies

Pros and cons

- Pro: Supports any operation and entity type
Entity access policies

Pros and cons

- Pro: Supports any operation and entity type
- Pro: Option to have an access overview UI
Entity access policies

Pros and cons

- Pro: Supports any operation and entity type
- Pro: Option to have an access overview UI
- Pro: Works alongside node grants (until hopefully removed in D9)
Entity access policies

Pros and cons

- Pro: Supports any operation and entity type
- Pro: Option to have an access overview UI
- Pro: Works alongside node grants (until hopefully removed in D9)
- Pro: Easy to work around a problematic module
Entity access policies

Pros and cons

- Pro: Supports any operation and entity type
- Pro: Option to have an access overview UI
- Pro: Works alongside node grants (until hopefully removed in D9)
- Pro: Easy to work around a problematic module
- Con: Too big of a change at once to go into core
Entity access policies

Pros and cons

- Pro: Supports any operation and entity type
- Pro: Option to have an access overview UI
- Pro: Works alongside node grants (until hopefully removed in D9)
- Pro: Easy to work around a problematic module
- Con: Too big of a change at once to go into core
- Con: Loads a list of config entities to decide access to another list of entities
Entity access policies

Pros and cons

- Pro: Supports any operation and entity type
- Pro: Option to have an access overview UI
- Pro: Works alongside node grants (until hopefully removed in D9)
- Pro: Easy to work around a problematic module
- Con: Too big of a change at once to go into core
- Con: Loads a list of config entities to decide access to another list of entities
- Con: Nowhere near finished
A query-level entity access API
A query-level entity access API

- Registered as an entity type handler
A query-level entity access API

- Registered as an entity type handler
- Generates conditions based on the user's permissions which are ultimately added to the query
A query-level entity access API

- Registered as an entity type handler
- Generates conditions based on the user's permissions which are ultimately added to the query
- Fires an event to alter the conditions (so before query alter)
A query-level entity access API

- Registered as an entity type handler
- Generates conditions based on the user's permissions which are ultimately added to the query
- Fires an event to alter the conditions (so before query alter)
- Works for Views and entity queries
A query-level entity access API

Pros and cons
A query-level entity access API

Pros and cons

- Pro: Familiar approach (entity handlers)
A query-level entity access API

Pros and cons

- Pro: Familiar approach (entity handlers)
- Pro: Works well for most scenarios out of the box
A query-level entity access API

Pros and cons

- Pro: Familiar approach (entity handlers)
- Pro: Works well for most scenarios out of the box
- Pro: Small enough in scope to make it into core
A query-level entity access API

Pros and cons

- Pro: Familiar approach (entity handlers)
- Pro: Works well for most scenarios out of the box
- Pro: Small enough in scope to make it into core
- Con: Relies on a permission handler that hasn't made it to core yet
A query-level entity access API

Pros and cons

- Pro: Familiar approach (entity handlers)
- Pro: Works well for most scenarios out of the box
- Pro: Small enough in scope to make it into core
- Con: Relies on a permission handler that hasn't made it to core yet
- Con: Hard to alter or prevent another module's access logic
Good news everyone
Work on query access handler to be committed to Entity API within a week
Permission handler on a fast-track to be added to core
https://www.drupal.org/node/2809177
This paves the way for adding the query access handlers to core
Likely in Drupal 8.7.0
Become a Drupal contributor
Friday from 9am

- First timers workshop
- Mentored contribution
- General contribution
Thank you!
Questions?