EMBRACE LEGACY

NEX 3 and Nikkor 50 1:8 © Bill Blyleven

PHIL GIESE

- With Signavio since 2009
- I'm a hacker at heart
- (fancy) Product Manager
- Still codes

WHAT IS LEGACY?

- Dead code (code that isn’t used in your application anymore)
- “Finished” code: features that change so little that they are not being re-factored in a long time
- Too complex code that is very hard to change and therefore isn’t changed
- Good, re-factored code that needs to change because you changed your coding style or changed something in the technology stack.

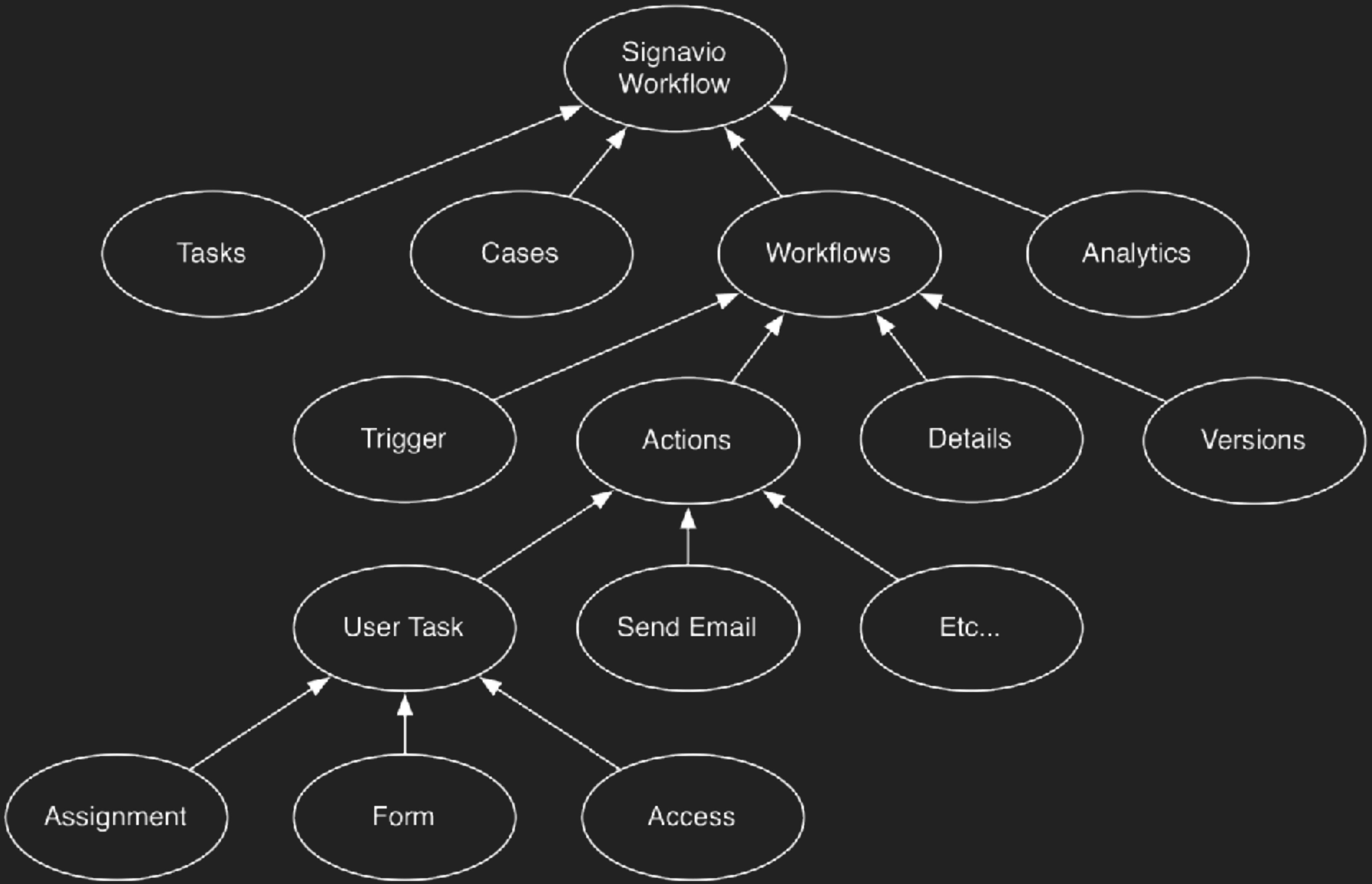
NOTHING IS EVER PERFECT

YOU WILL NEED TO REFACTOR

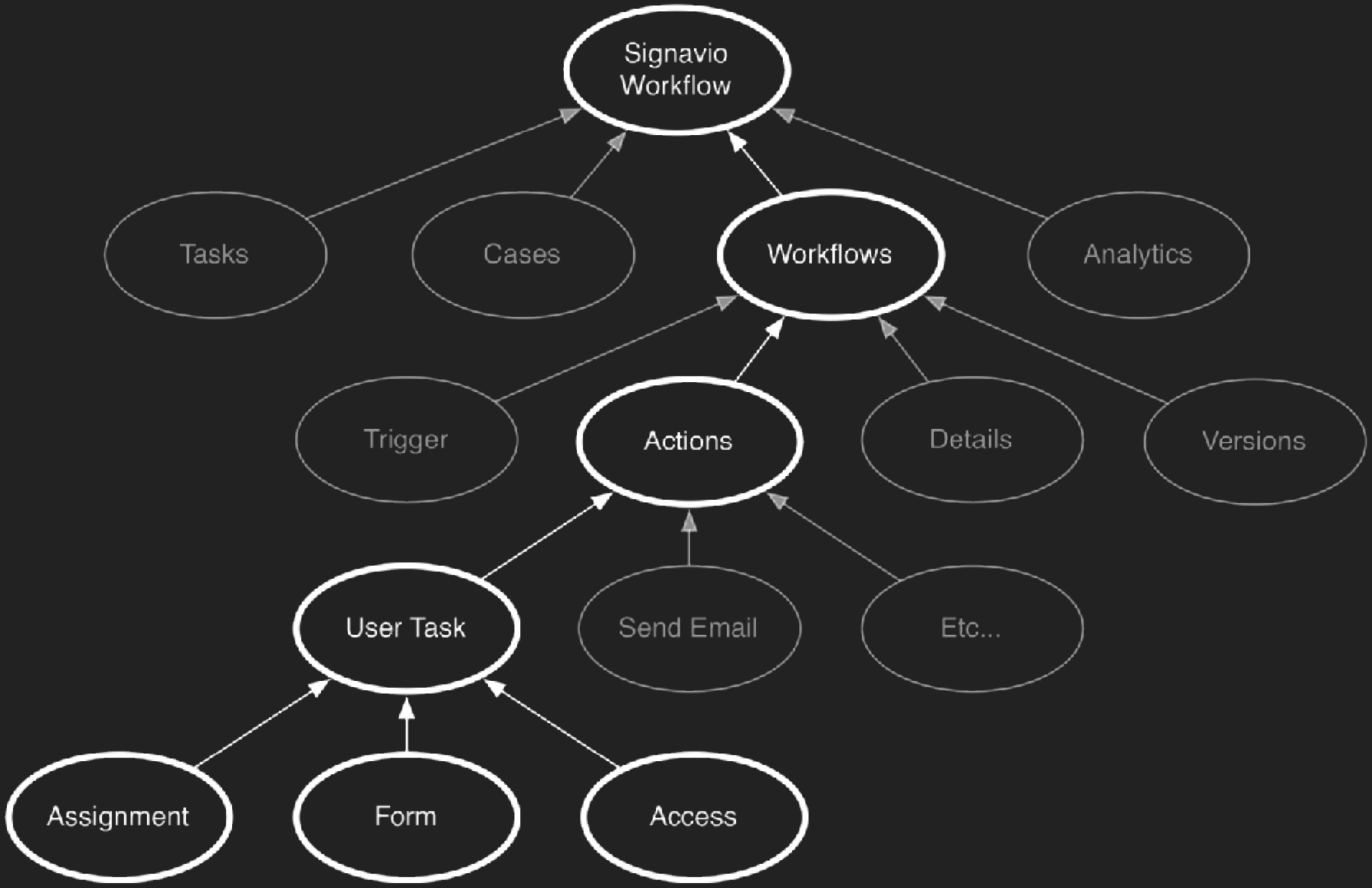
WHAT THIS TALK IS ACTUALLY ABOUT

- Big technology changes
- Changes that take a long time to finish
- How not to piss off everyone else
- Scale with cool

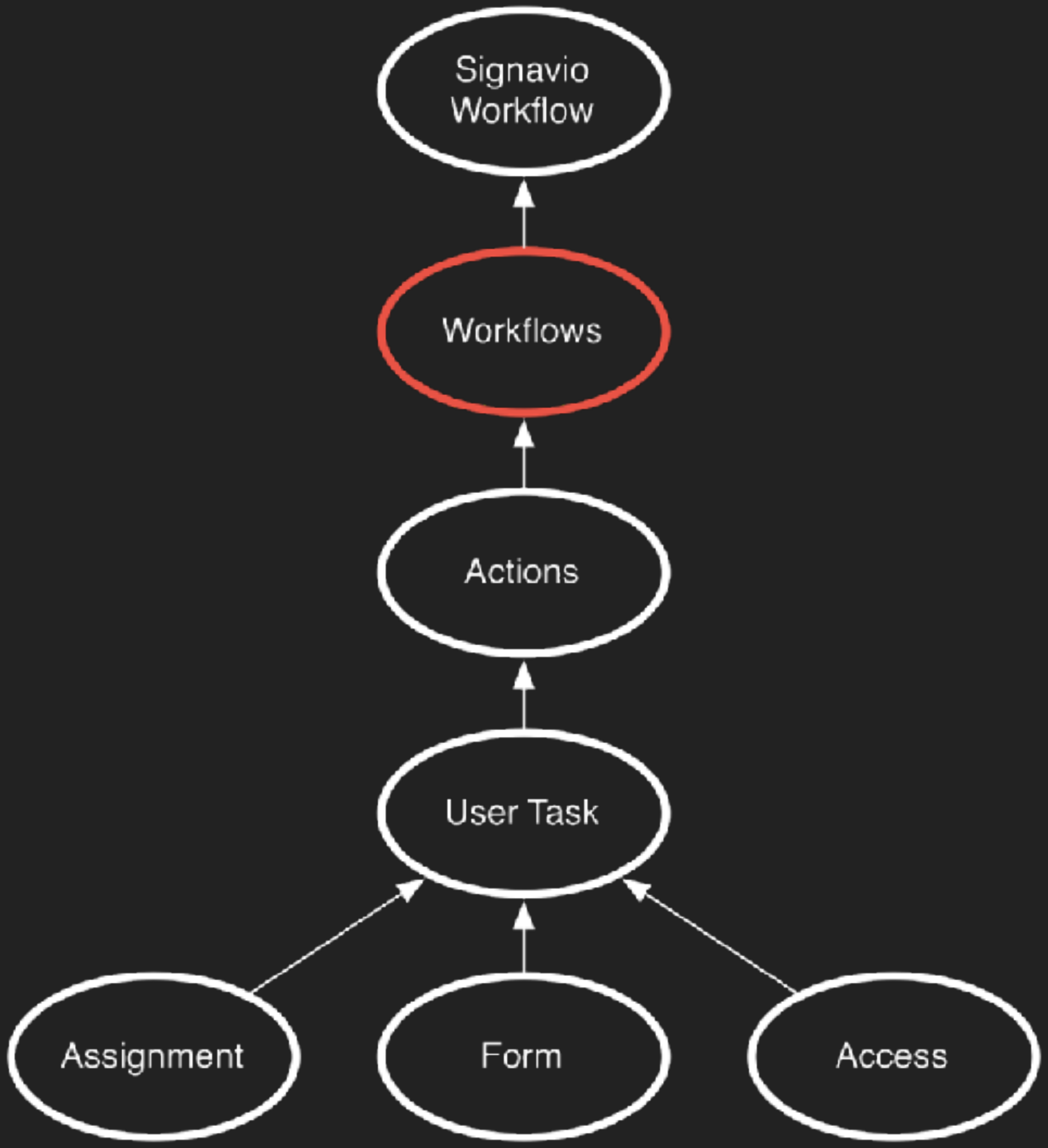
THIS IS HOW YOUR APPLICATION LOOKS LIKE



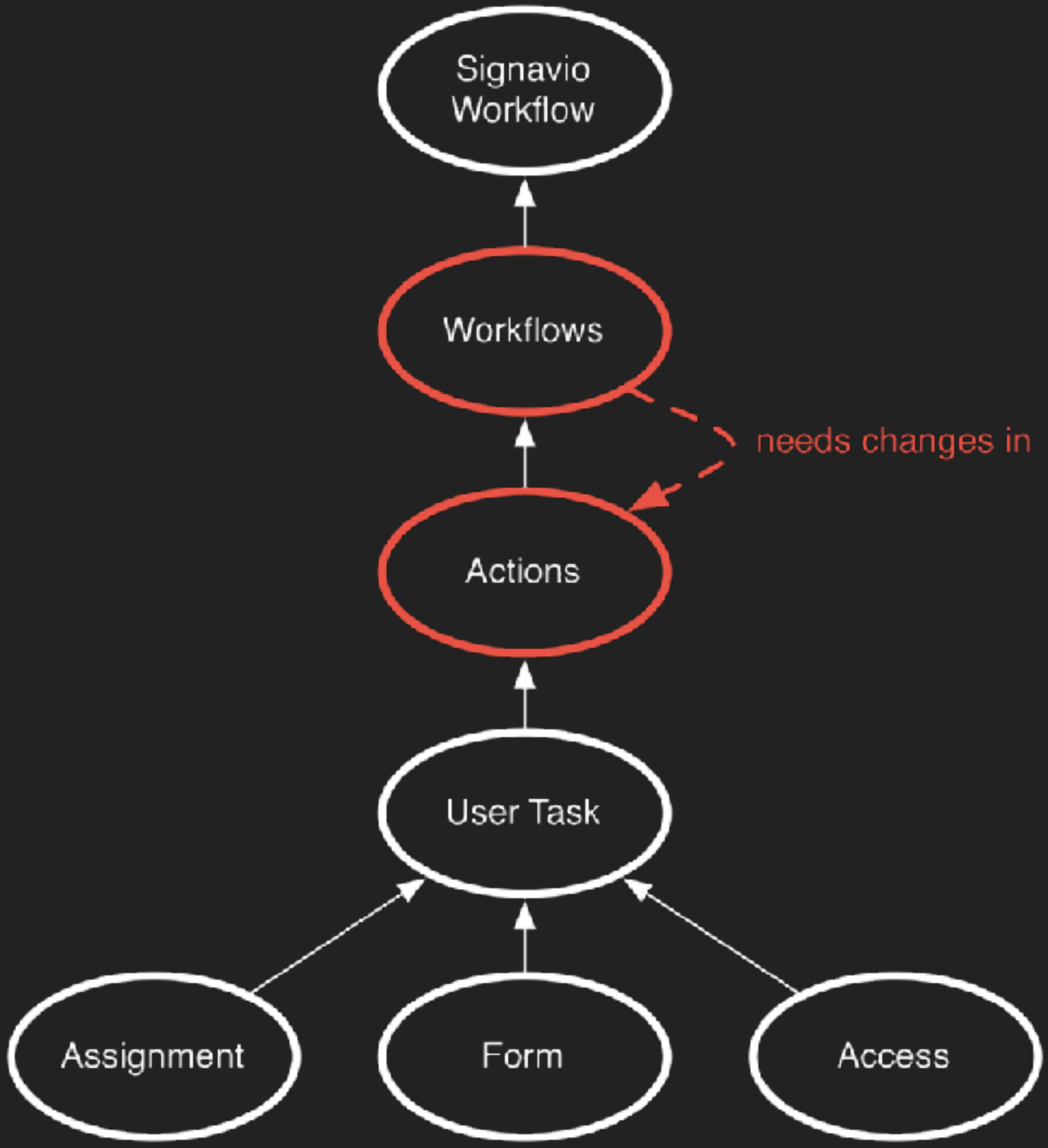
THIS IS HOW YOUR APPLICATION LOOKS LIKE



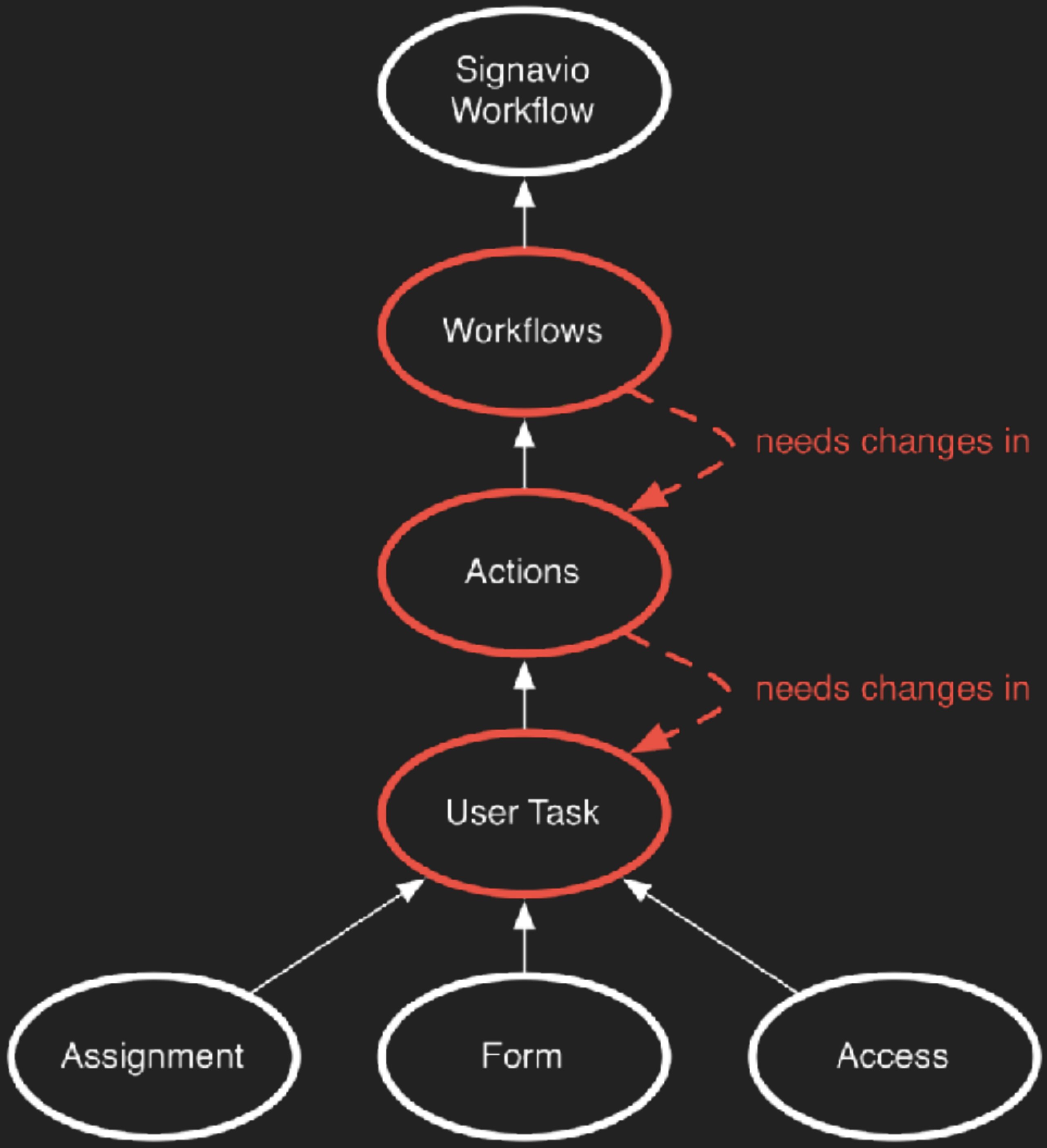
HOW NOT TO DO IT



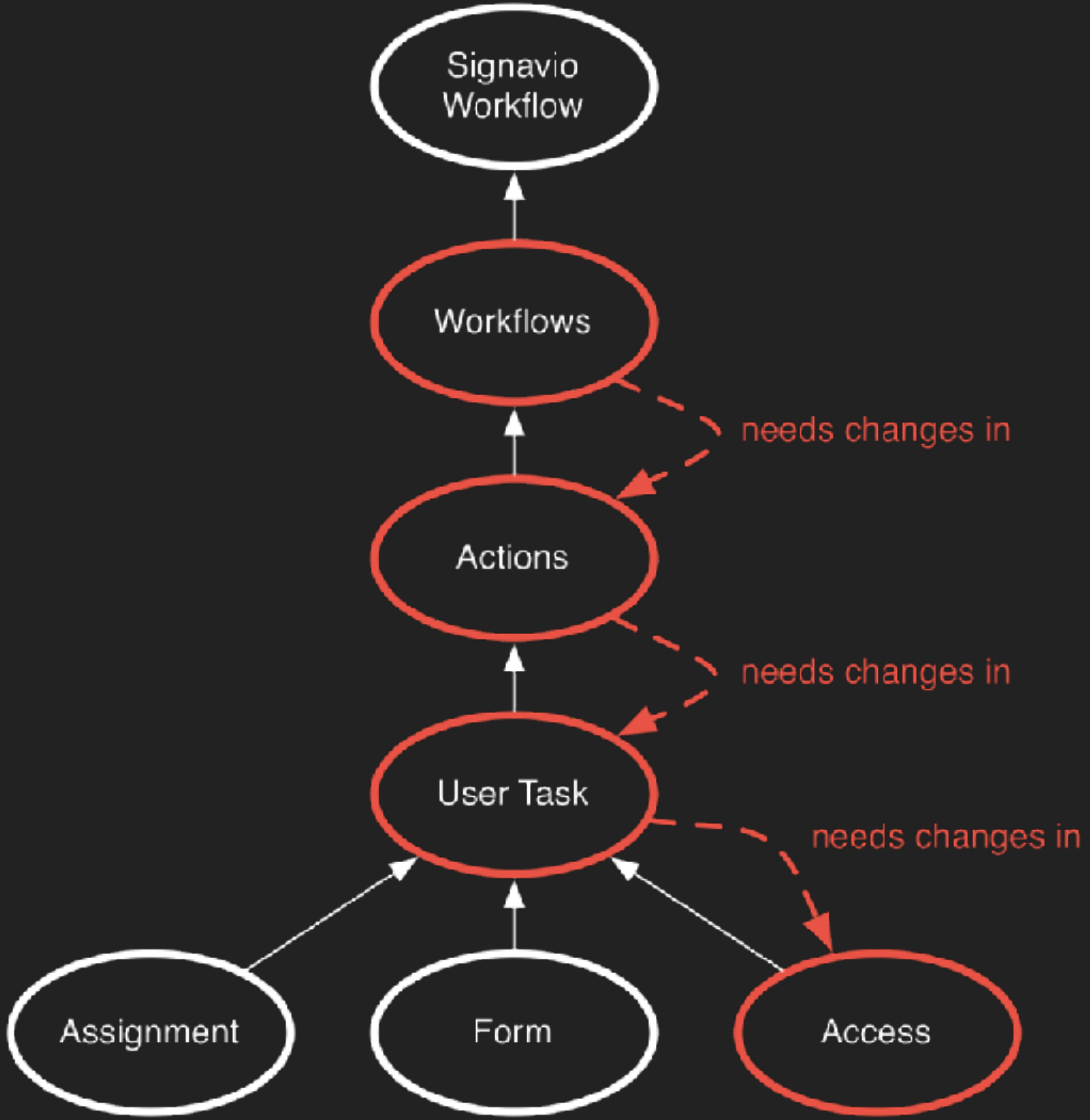
HOW NOT TO DO IT



HOW NOT TO DO IT

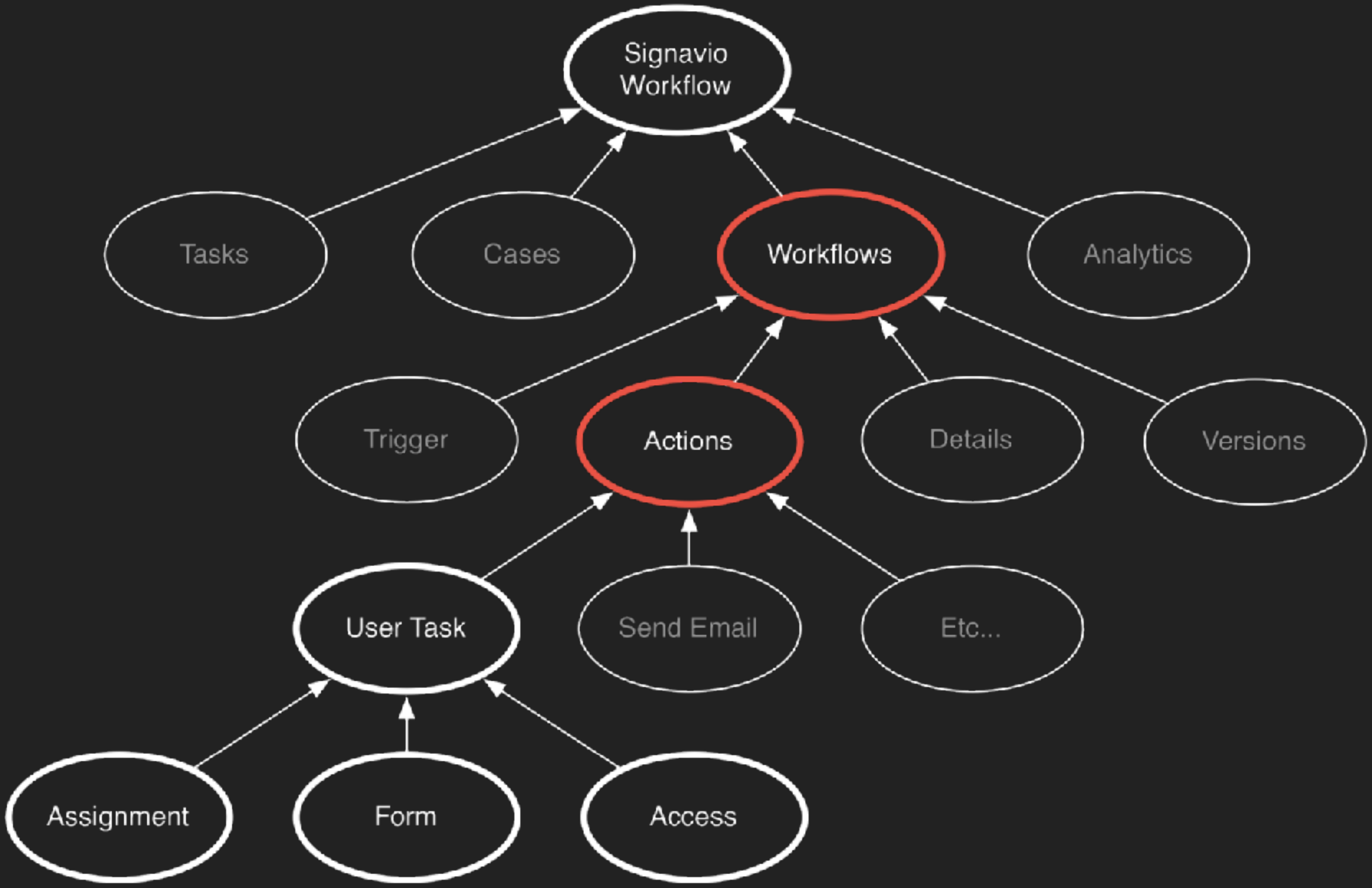


HOW NOT TO DO IT

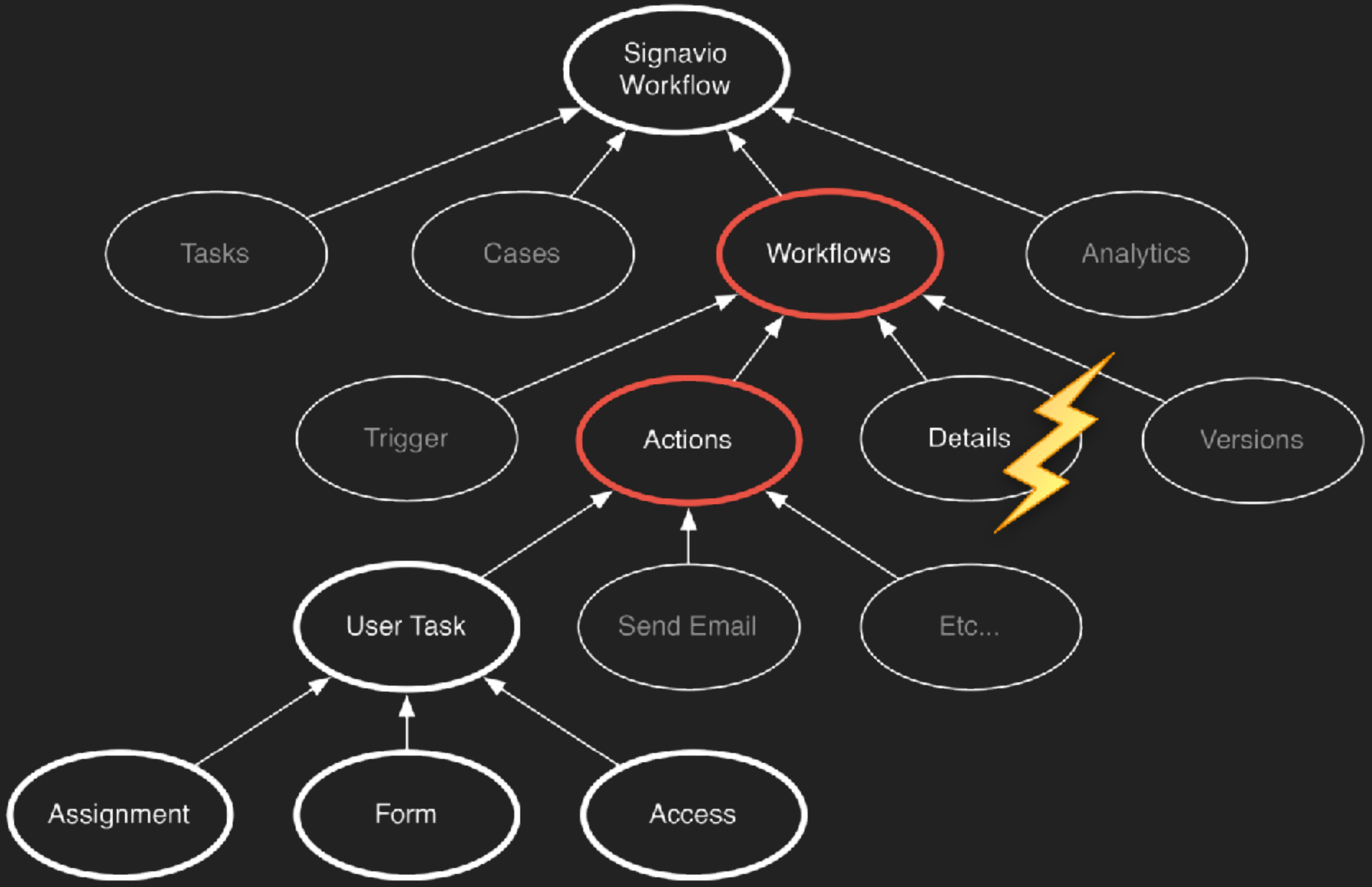


THIS IS A PROBLEM WHY?

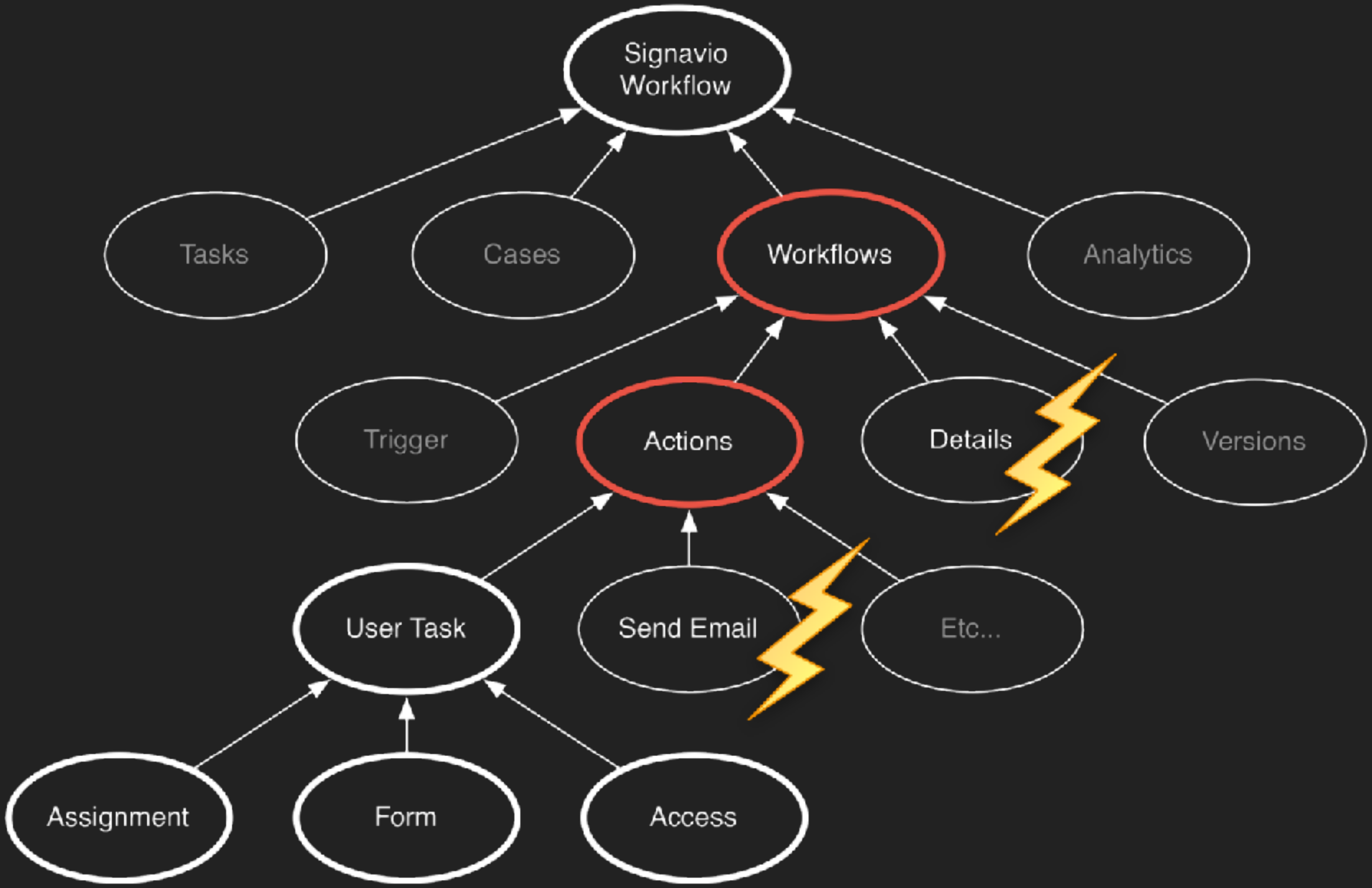
THIS IS HOW YOUR APPLICATION LOOKS LIKE



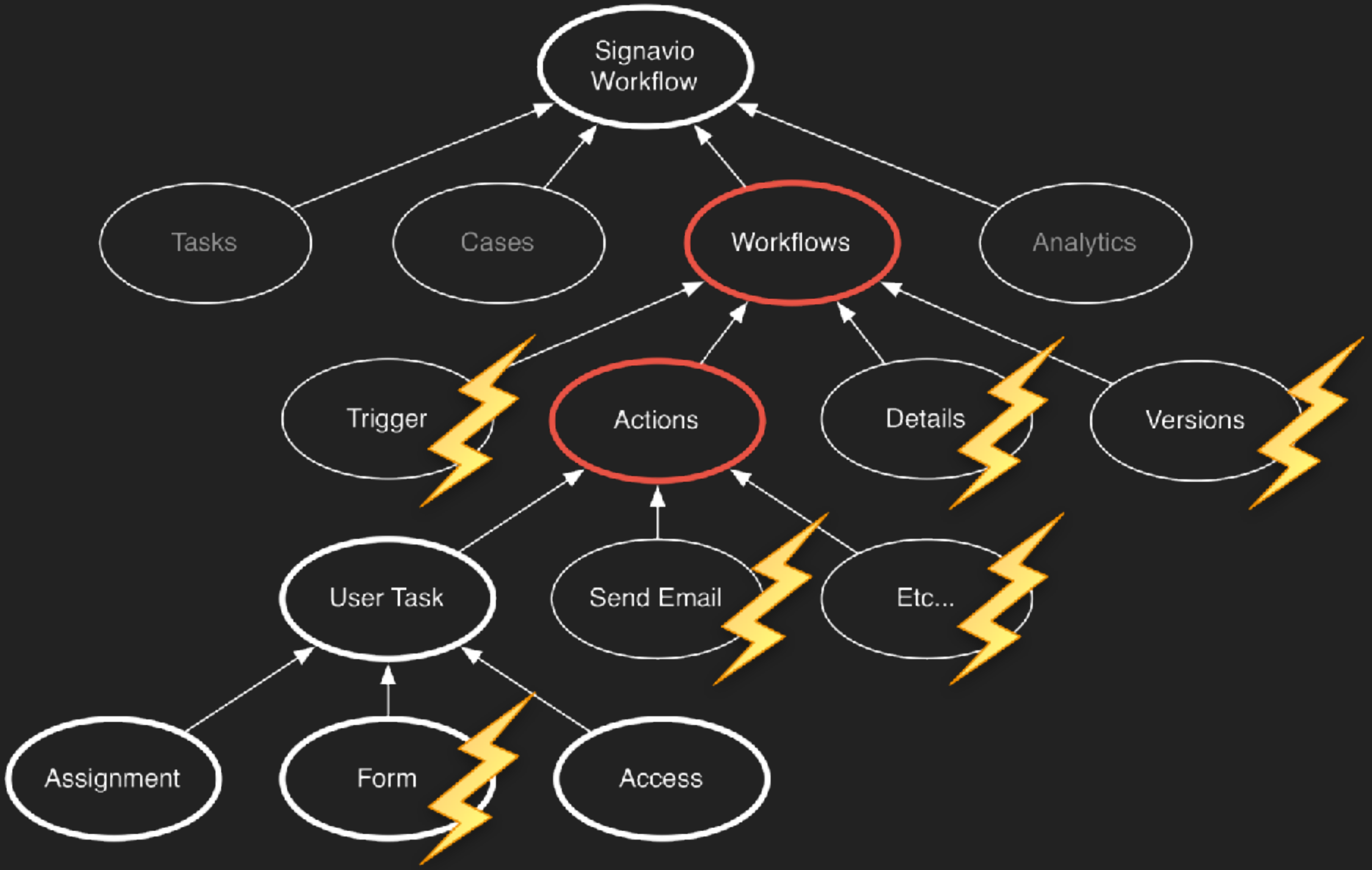
THIS IS HOW YOUR APPLICATION LOOKS LIKE



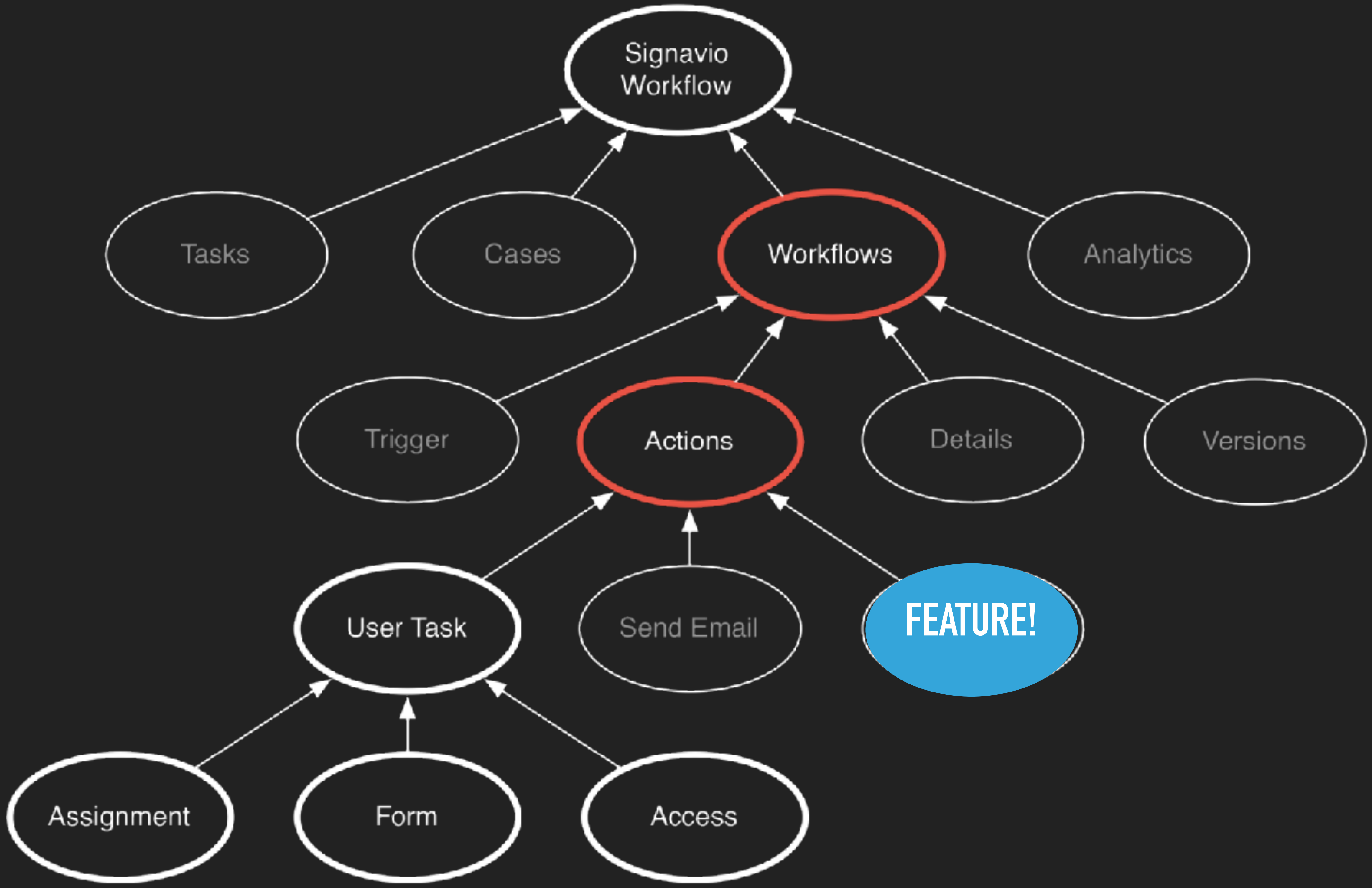
THIS IS HOW YOUR APPLICATION LOOKS LIKE



THIS IS HOW YOUR APPLICATION LOOKS LIKE



THIS IS HOW YOUR APPLICATION LOOKS LIKE



SIGNAVIO
Philipp Giese

PEOPLE WILL HATE YOU

WHY OTHER DEVELOPERS HATE YOU

- Your big refactoring adds tons of conflicts
- They need to fix those
- What they have built does not work anymore because of you
- They now work for you

WHY PRODUCT MANAGEMENT HATES YOU

- You keep everyone busy but with no obvious result
- New features are not being added
- Existing features break
- Planning is impossible

WHY SUPPORT HATES YOU

- Every time you ship, you destroy what was already there
- Customers get frustrated because things that used to work break and new features aren’t delivered to compensate for that

NO MORE FUN FOR YOU

WHAT HAPPENED?

YOUR ASSUMPTIONS ARE WRONG

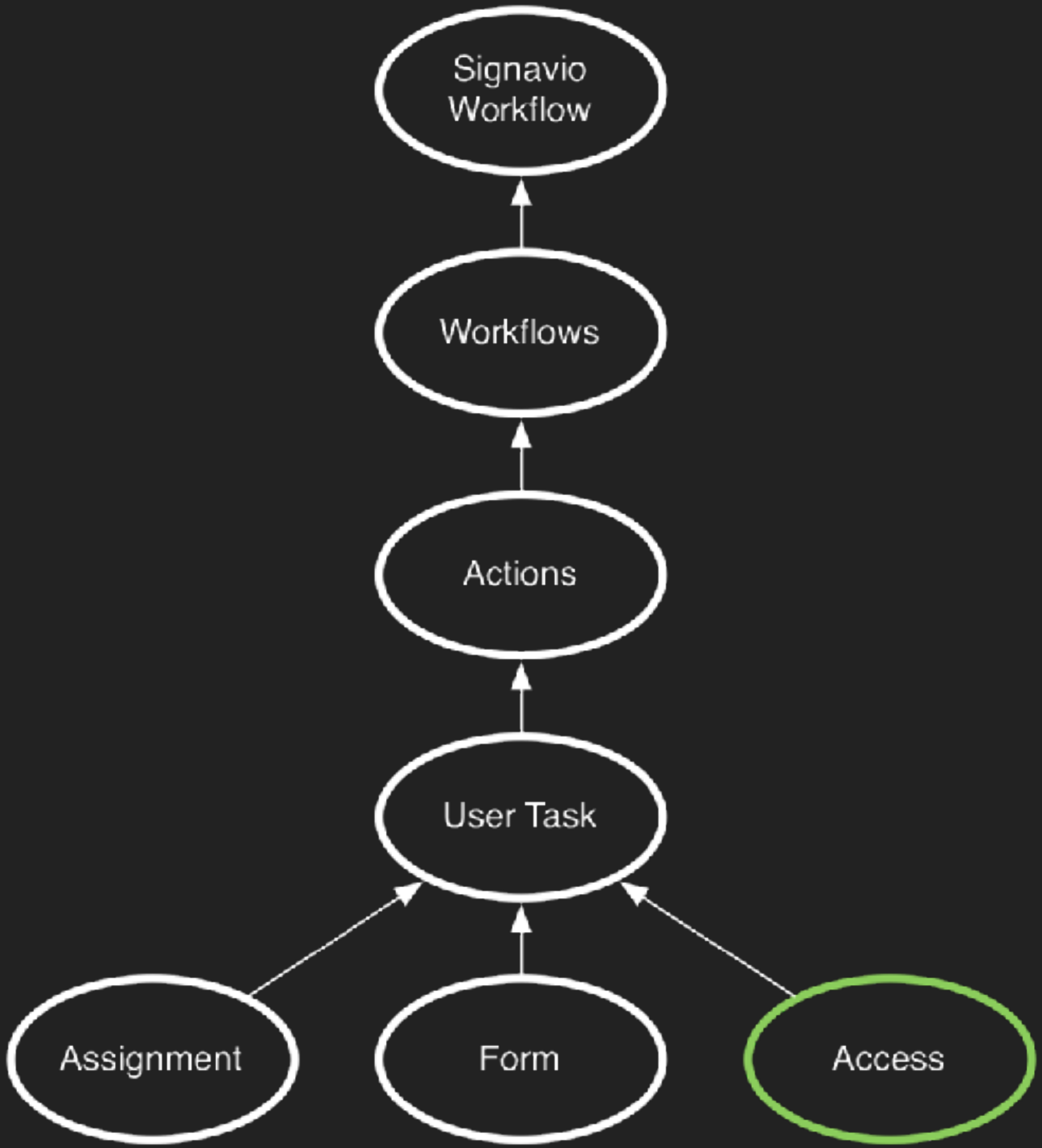
- The effects of your change aren't that local at all
- The deeper you get into the rabbit hole the darker it gets
- What you thought would be a small easy change blows up into your face
- Since you block everything at top level you can only merge back to master when you're done

YOU SCREWED UP, WHAT NOW?

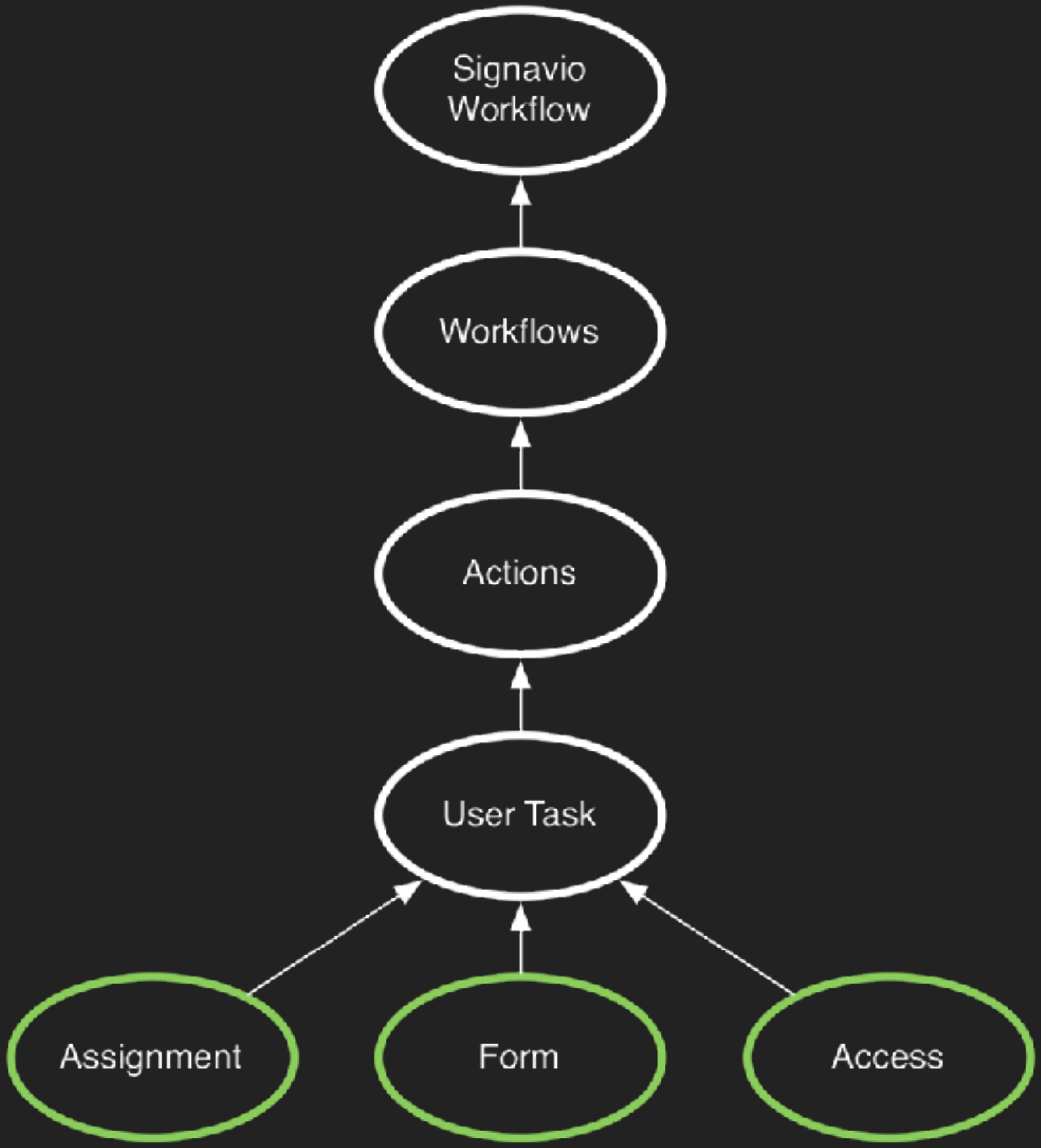
- Stop
- Think
- Revert
- Think
- Start again

OK, THEN HOW?

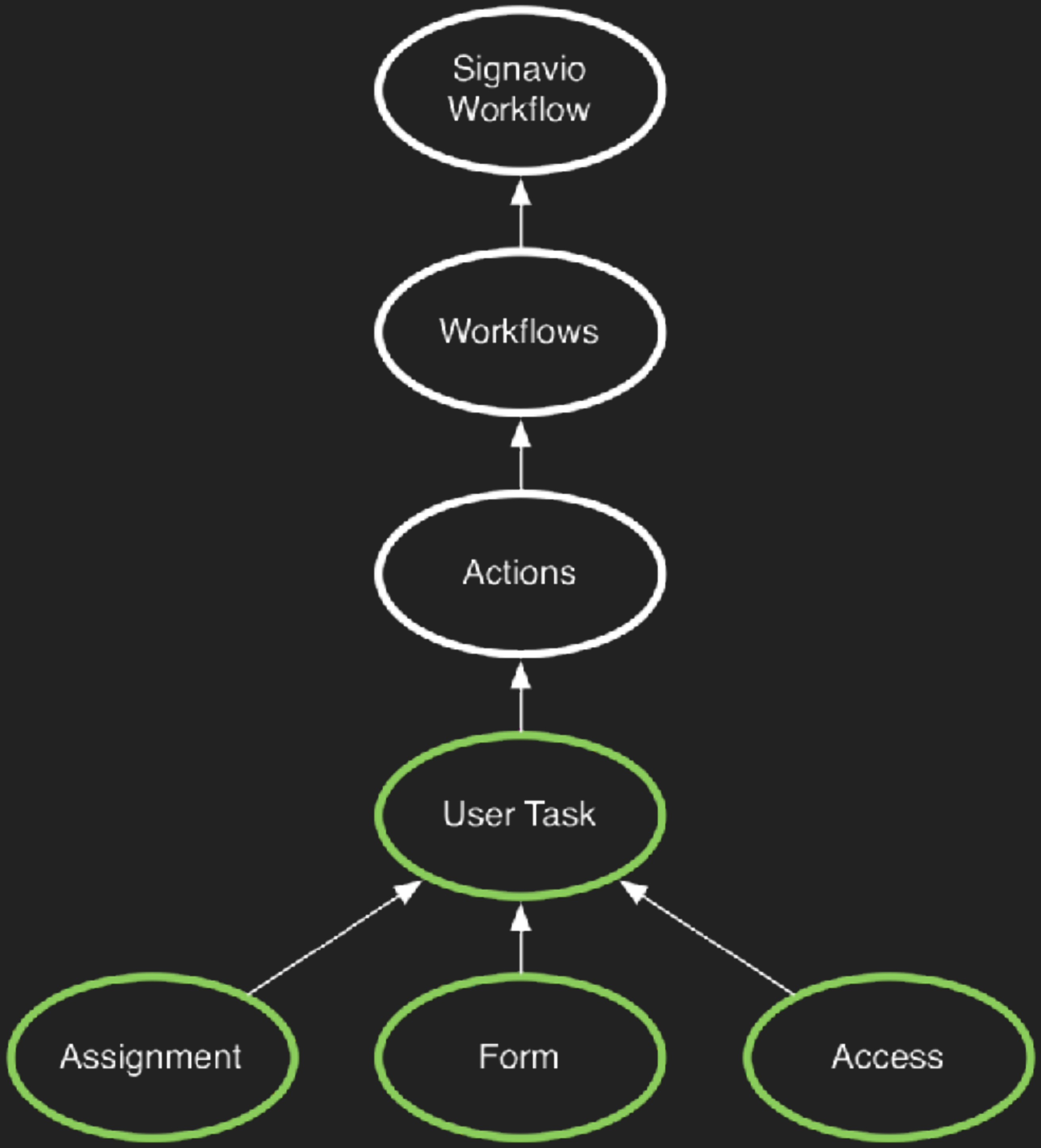
A NEW HOPE


Signavio Workflow
Workflows
Actions
User Task
Assignment
Form
Access

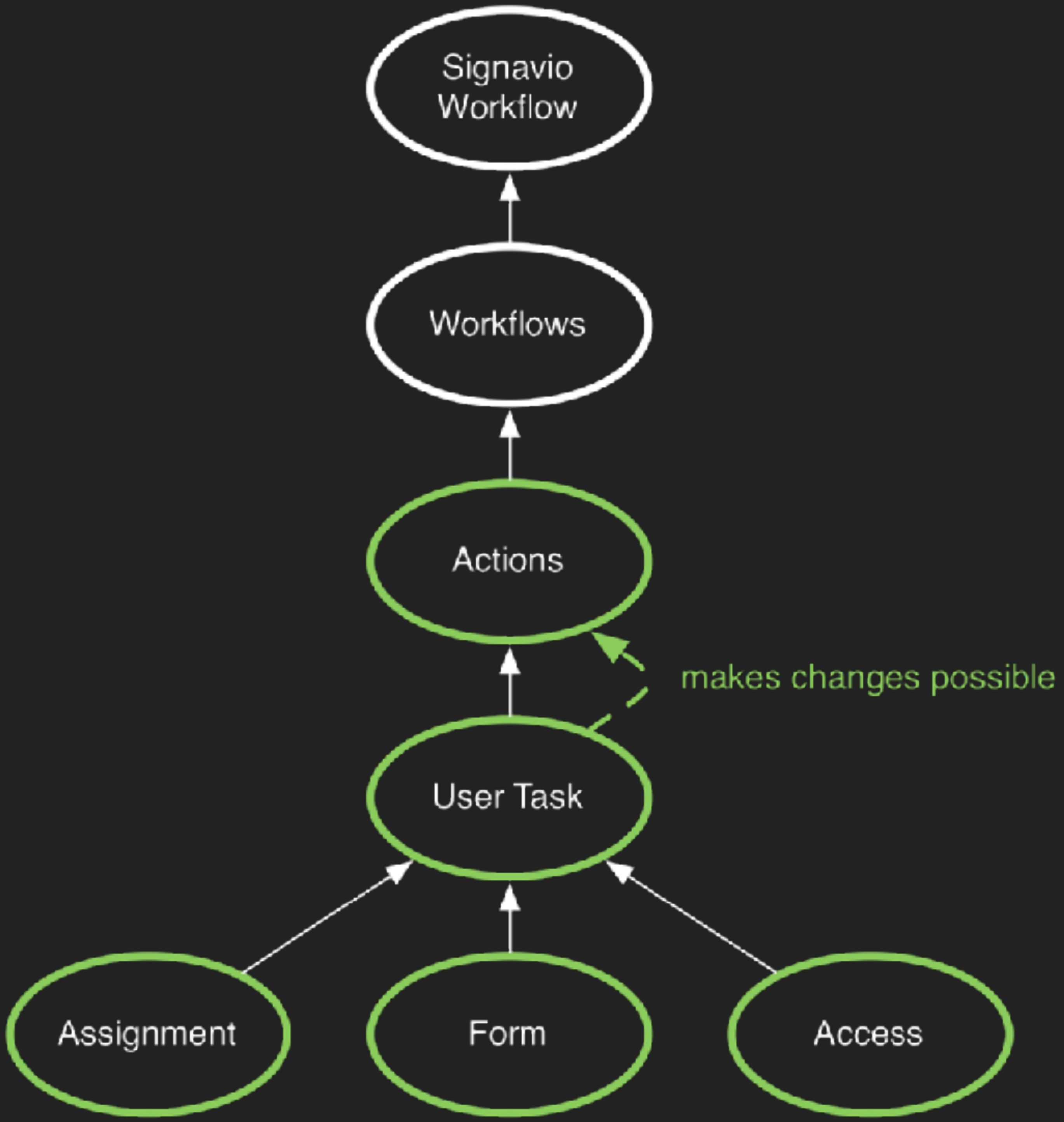
A NEW HOPE


Signavio Workflow
Workflows
Actions
User Task
Assignment
Form
Access

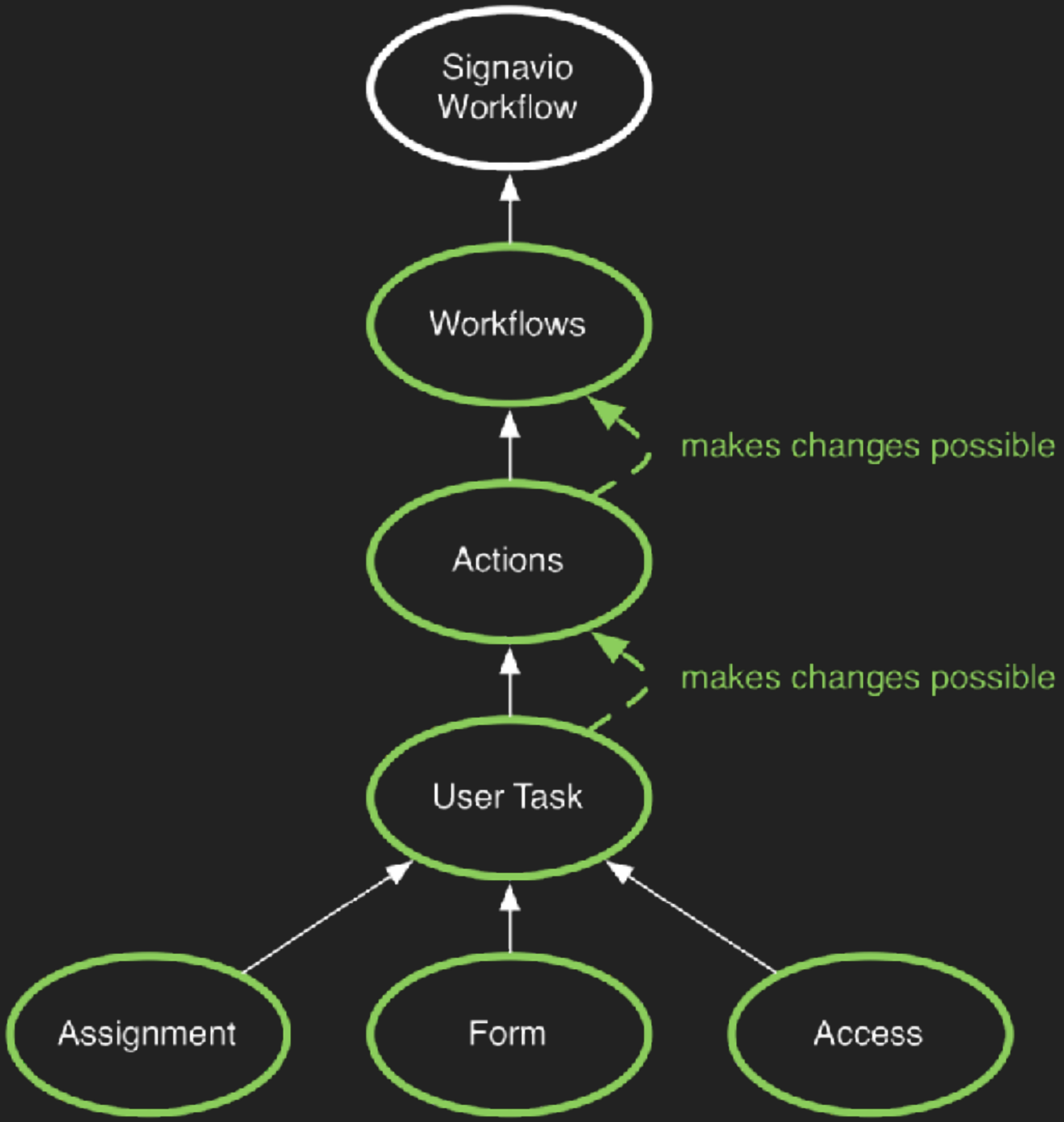
A NEW HOPE


Signavio Workflow
Workflows
Actions
User Task
Assignment
Form
Access

A NEW HOPE


Signavio Workflow
Workflows
Actions
makes changes possible
User Task
Assignment
Form
Access

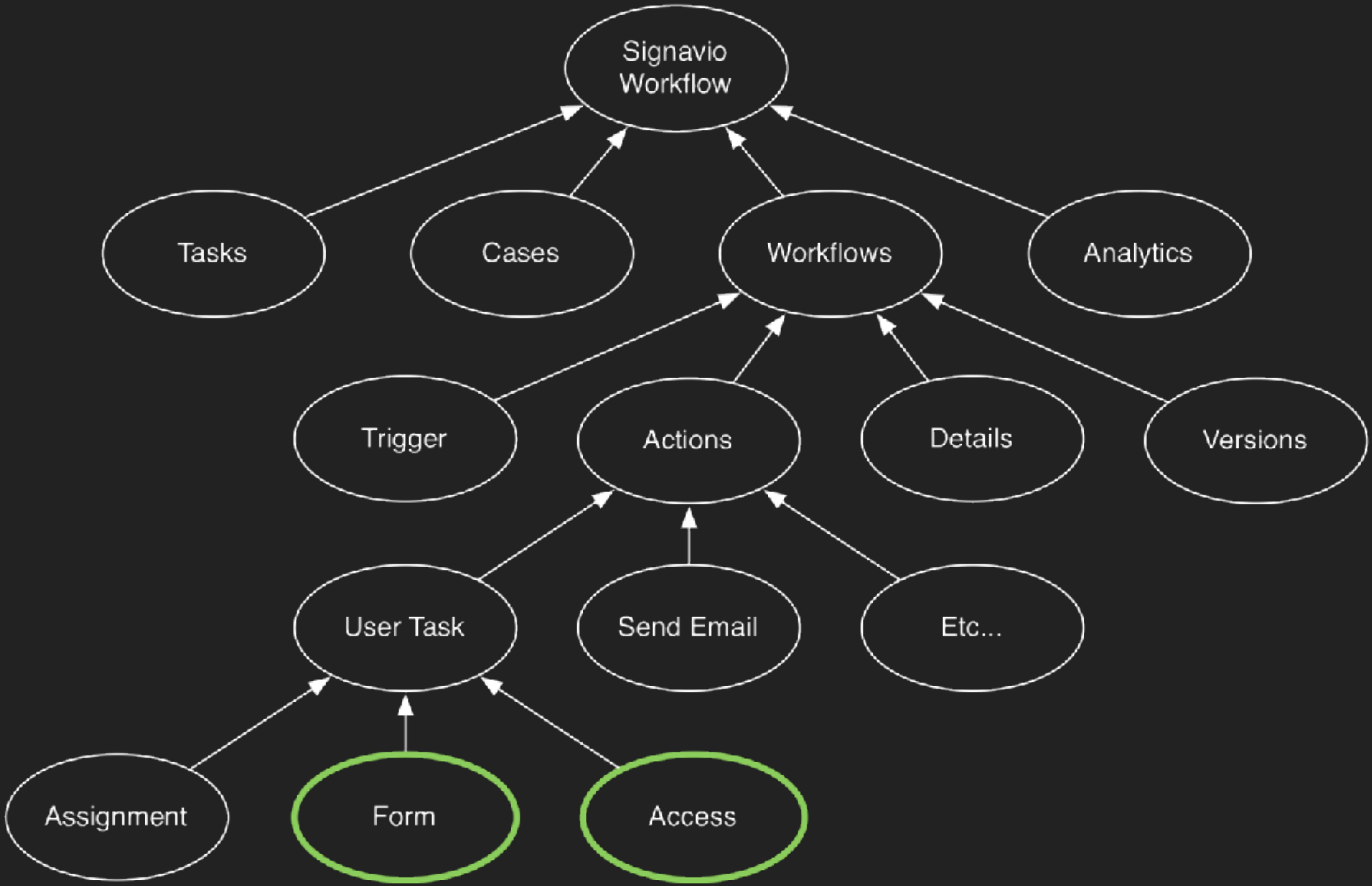
A NEW HOPE


Signavio Workflow
Workflows
makes changes possible
Actions
makes changes possible
User Task
Assignment
Form
Access

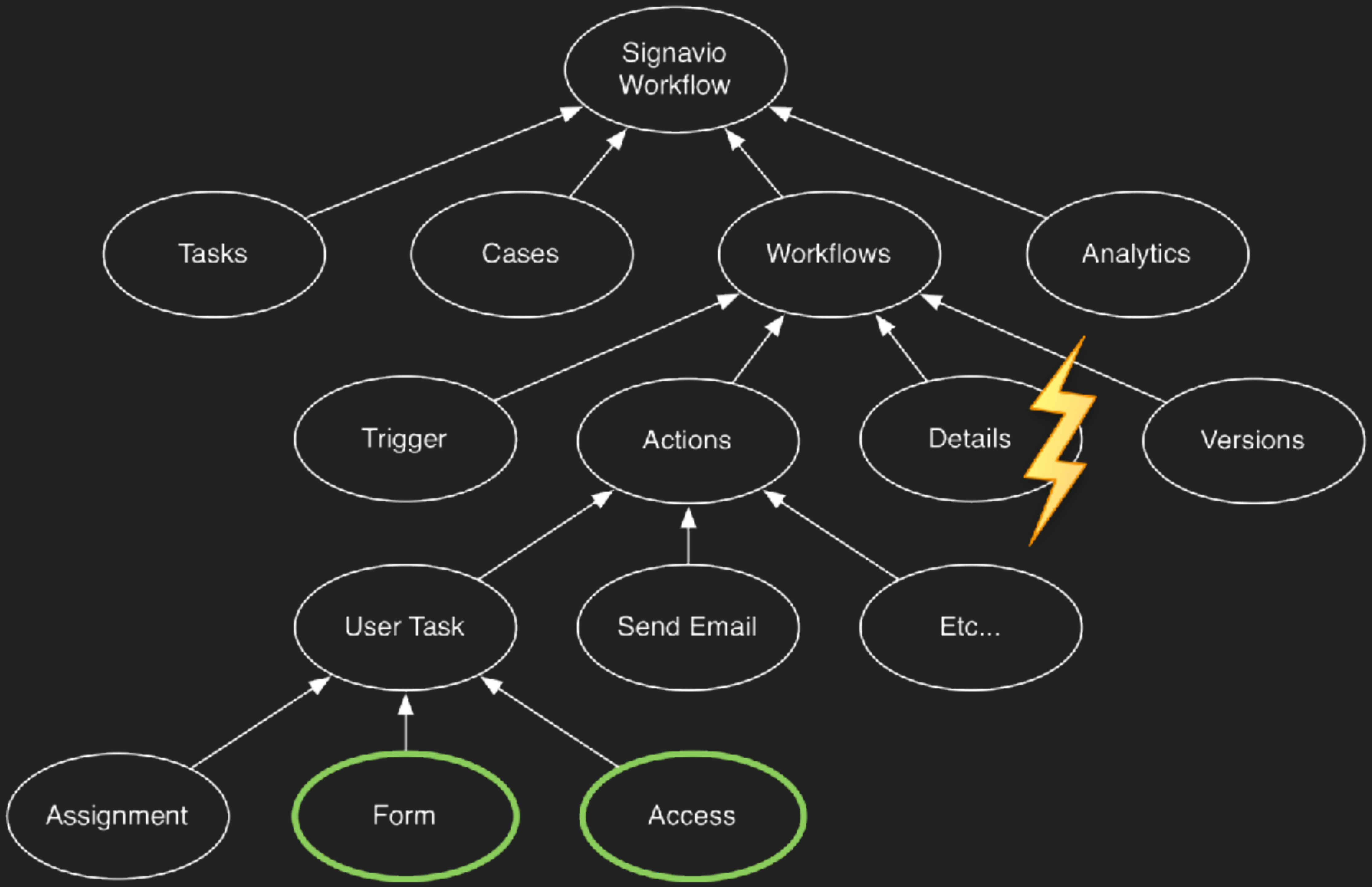
THIS IS BETTER
WHY?

ITERATIVE ROCKS!

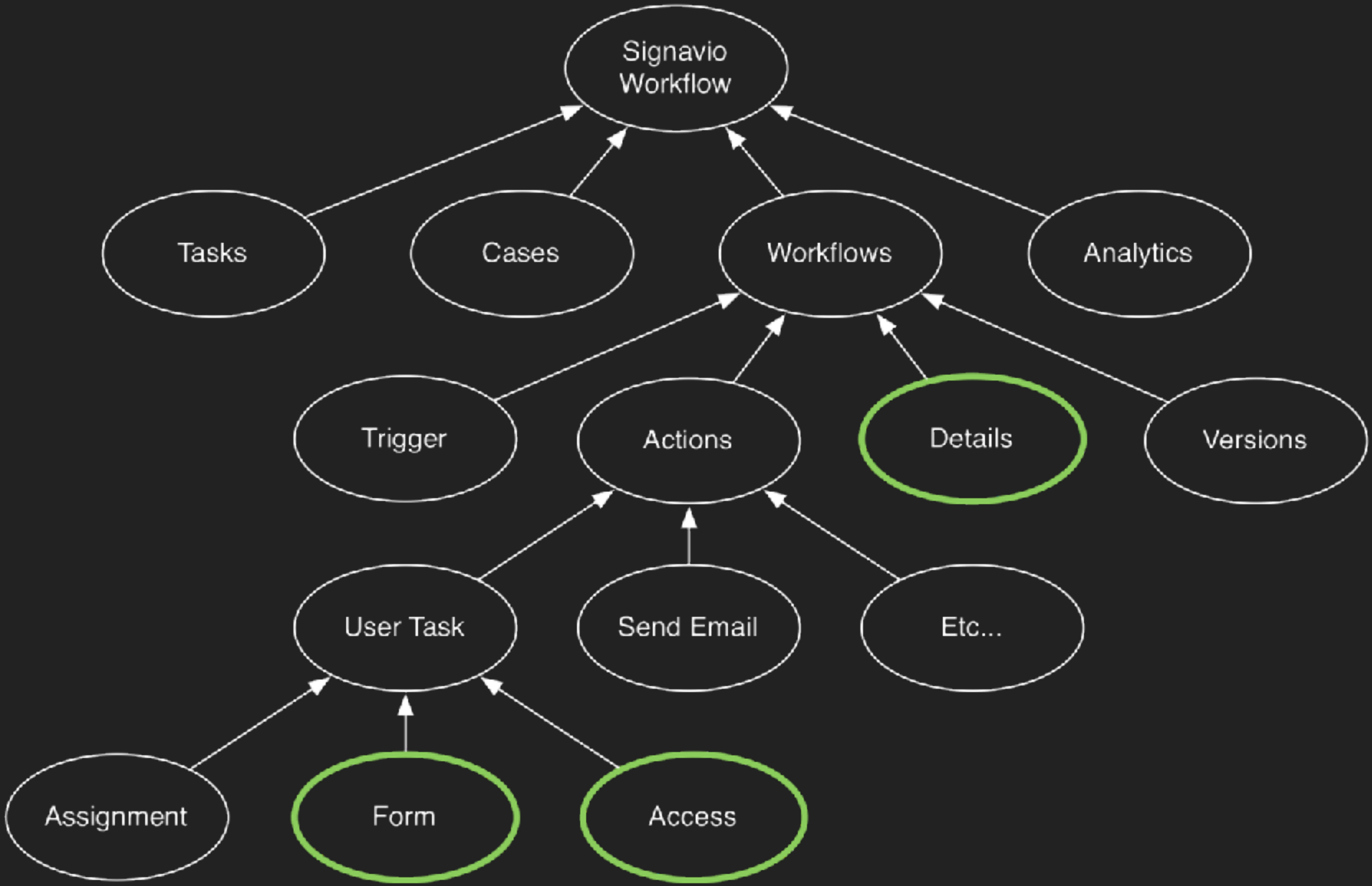
THIS IS HOW YOUR APPLICATION LOOKS LIKE



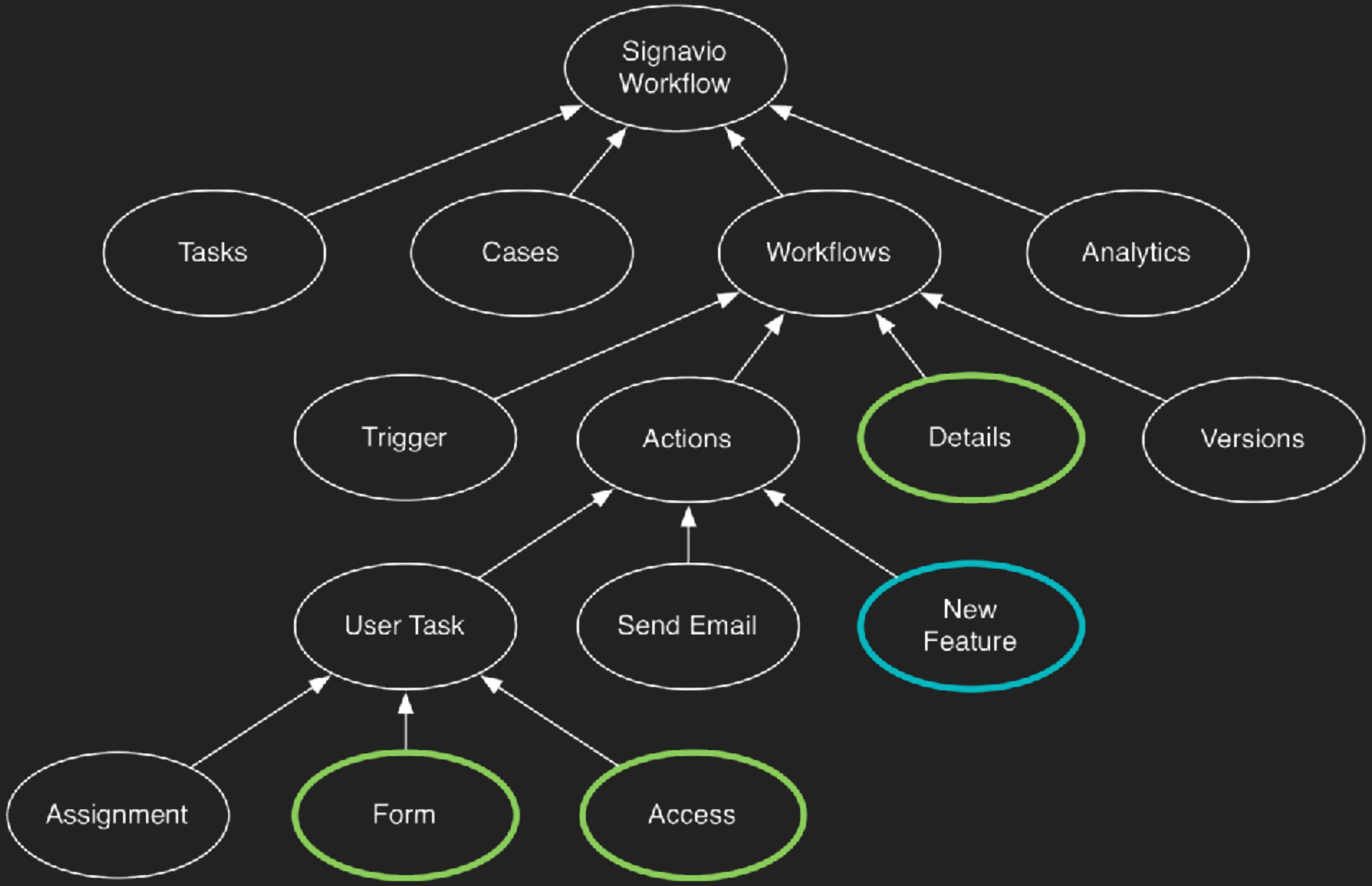
THIS IS HOW YOUR APPLICATION LOOKS LIKE



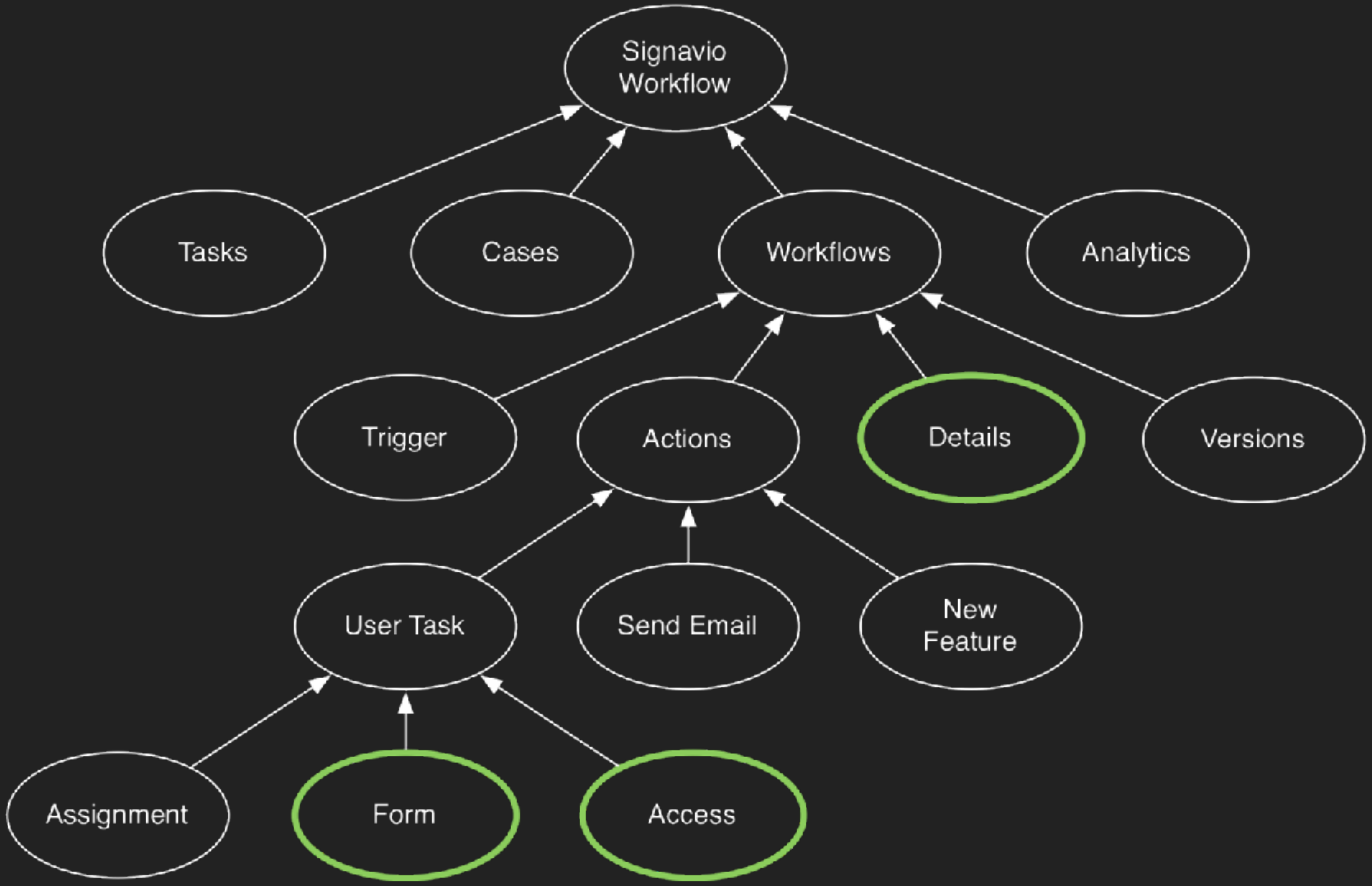
THIS IS HOW YOUR APPLICATION LOOKS LIKE



THIS IS HOW YOUR APPLICATION LOOKS LIKE



THIS IS HOW YOUR APPLICATION LOOKS LIKE

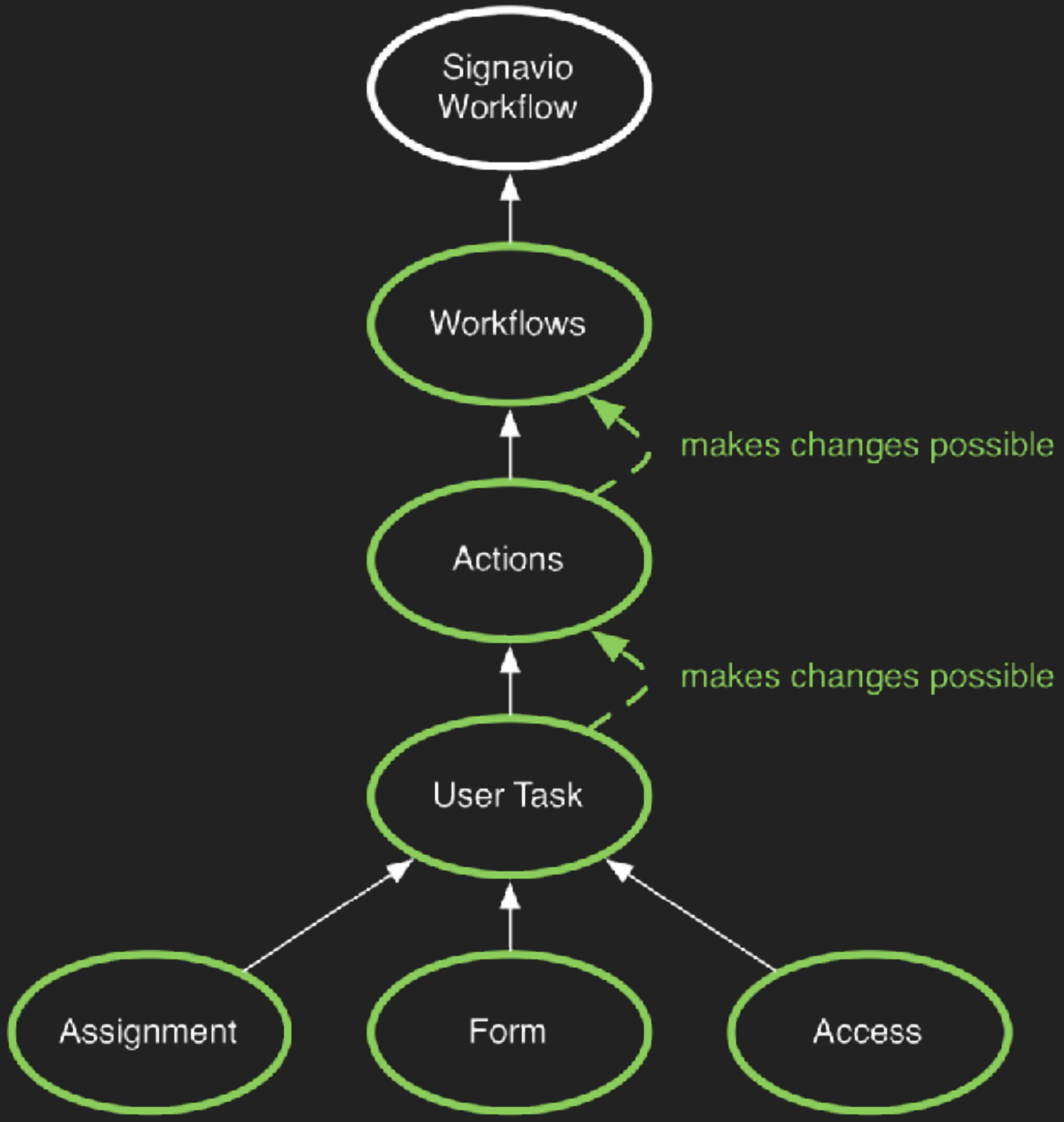


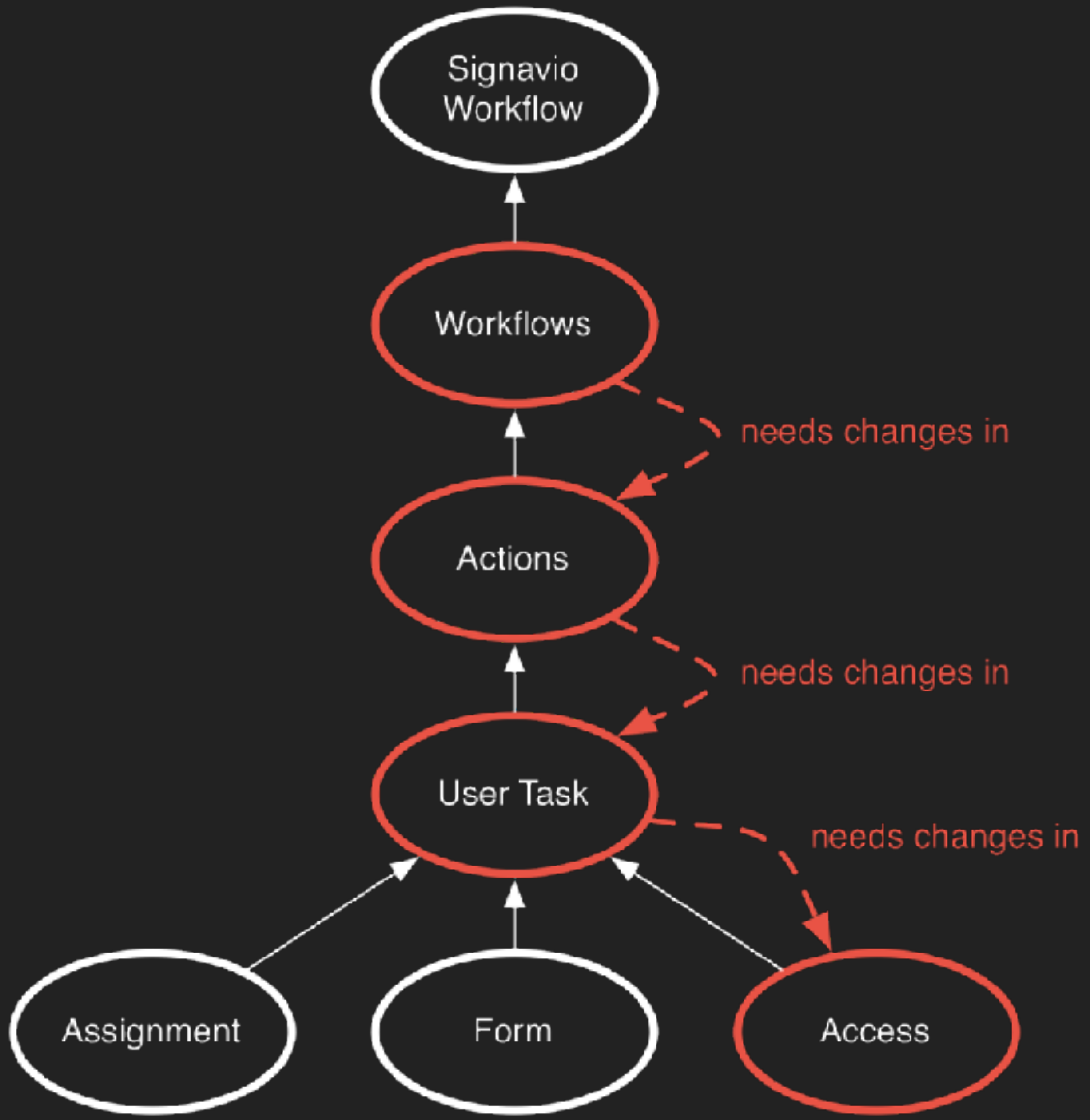
WORK ITERATIVELY, BE READY FOR THE NEXT CHANGE

- Every change is isolated and has minimal effect
- Changes on other nodes along the way do not affect you
- You can merge your changes back to master after every node
- If unforeseen things happen, their impact stays local to what you are currently doing

RECAP

SMALL ITERATIVE CHANGE OVER BIG, BLOCKING REFACTORING





FEEDBACK?

FURTHER READING

- https://www.youtube.com/watch?v=BF58ZJ1ZQxY
- https://mikadomethod.wordpress.com/
- https://pragprog.com/magazines/2010-06/the-mikado-method