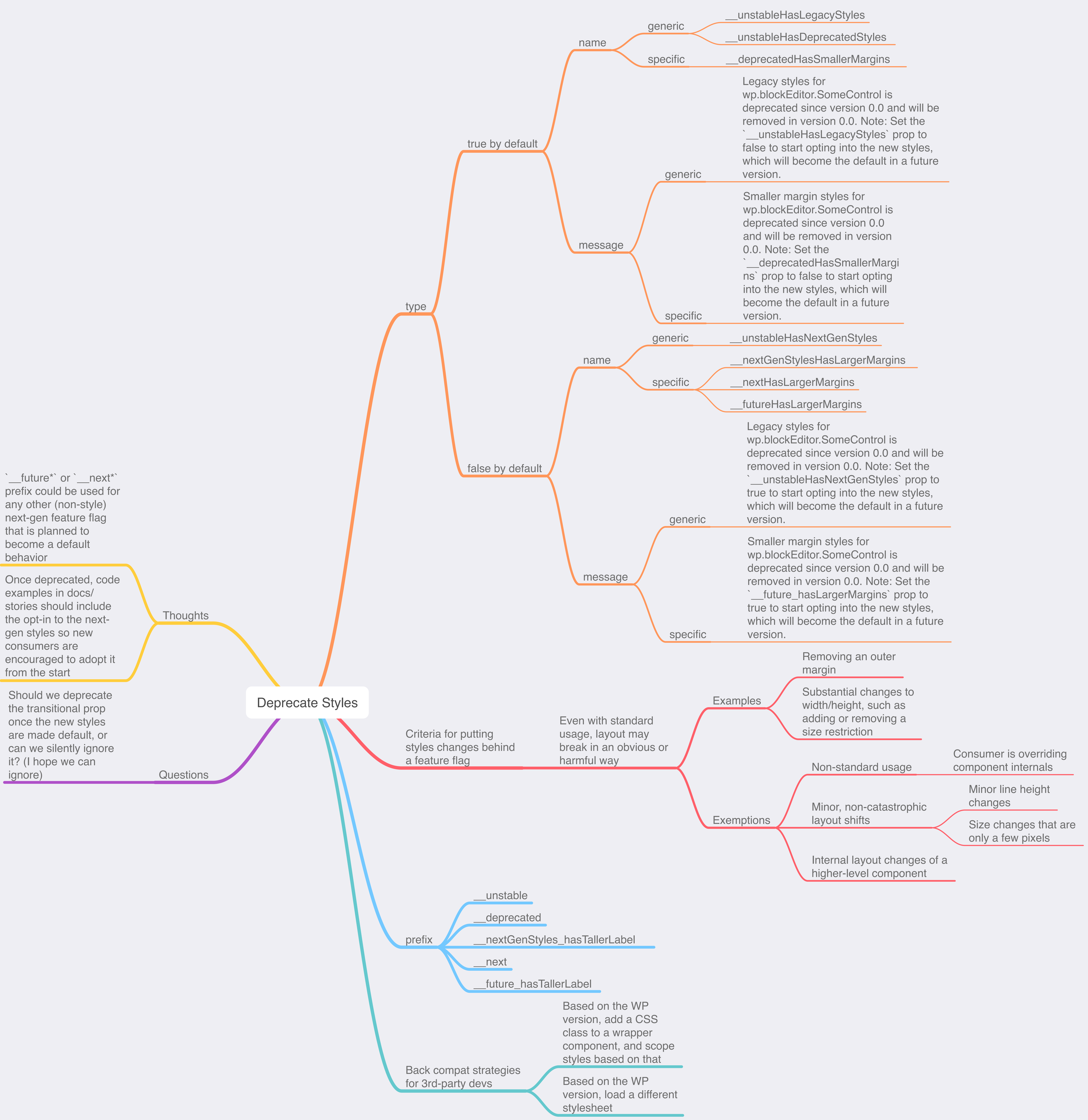

- Deprecate Styles
  - type
    - true by default
      - name
        - generic
          - __unstableHasLegacyStyles
          - __unstableHasDeprecatedStyles
        - specific
          - __deprecatedHasSmallerMargins
      - message
        - generic
          - Legacy styles for wp.blockEditor.SomeControl is deprecated since version 0.0 and will be removed in version 0.0. Note: Set the `__unstableHasLegacyStyles` prop to false to start opting into the new styles, which will become the default in a future version.
        - specific
          - Smaller margin styles for wp.blockEditor.SomeControl is deprecated since version 0.0 and will be removed in version 0.0. Note: Set the `__deprecatedHasSmallerMargins` prop to false to start opting into the new styles, which will become the default in a future version.
    - false by default
      - name
        - generic
          - __unstableHasNextGenStyles
        - specific
          - __nextGenStylesHasLargerMargins
          - __nextHasLargerMargins
          - __futureHasLargerMargins
      - message
        - generic
          - Legacy styles for wp.blockEditor.SomeControl is deprecated since version 0.0 and will be removed in version 0.0. Note: Set the `__unstableHasNextGenStyles` prop to true to start opting into the new styles, which will become the default in a future version.
        - specific
          - Smaller margin styles for wp.blockEditor.SomeControl is deprecated since version 0.0 and will be removed in version 0.0. Note: Set the `__future_hasLargerMargins` prop to true to start opting into the new styles, which will become the default in a future version.
  - Criteria for putting styles changes behind a feature flag
    - Even with standard usage, layout may break in an obvious or harmful way
      - Examples
        - Removing an outer margin
        - Substantial changes to width/height, such as adding or removing a size restriction
      - Exemptions
        - Non-standard usage
          - Consumer is overriding component internals
        - Minor, non-catastrophic layout shifts
          - Minor line height changes
          - Size changes that are only a few pixels
        - Internal layout changes of a higher-level component
  - prefix
    - __unstable
    - __deprecated
    - __nextGenStyles_hasTallerLabel
    - __next
    - __future_hasTallerLabel
  - Back compat strategies for 3rd-party devs
    - Based on the WP version, add a CSS class to a wrapper component, and scope styles based on that
    - Based on the WP version, load a different stylesheet
  - Thoughts
    - `__future*` or `__next*` prefix could be used for any other (non-style) next-gen feature flag that is planned to become a default behavior
    - Once deprecated, code examples in docs/ stories should include the opt-in to the next-gen styles so new consumers are encouraged to adopt it from the start
  - Questions
    - Should we deprecate the transitional prop once the new styles are made default, or can we silently ignore it? (I hope we can ignore)