XKB configuration: comparing XFree 4.3.0 and 'classic' configuration schemes

The XKB protocol: general information and criticism

The X keyboard extension for the X protocol (XKB) was developed by SGI and standardized by the X Consortium around 1996. This extension was intended to resolve the problems with keyboard handling presented in the core X protocol. In other words, “The X Keyboard Extension provides capabilities that are lacking or are cumbersome in the core X protocol”[1].

Much differing terminology is used throughout the XKB documentation, however in the context of this document, the most important concepts are 'groups' and 'layouts'.

A 'group' is a logical state of a keyboard, providing access to a collection of characters. Usually these characters belong together and may be arranged on several levels within a group (for simplicity, a 'level' can be considered as the state set by Shift/CapsLock/etc modifier keys).

Layouts usually express national variations (eg. German) but they can also be used for alternate keyboard layouts (eg. Dvorak) or other such oddities. Any layout can describe one (eg. French) or several (eg. Russian) groups. Each group within a layout has unique name. A layout can optionally have several variants which represent small changes within the groups.

The XKB protocol and API resolved most of the basic problems with keyboard handling within the X environment, and were a huge step forward (especially for users dealing with national keyboards). However the XKB architecture has its own limitations which have proved annoying to many users worldwide. The most significant usability issue caused by the protocol itself is the fixed maximum number of 4 groups which can be used at once without XKB reconfiguration, due to the fact that there are only 2 bits allocated in the protocol packets for the group number. Another weak point – common for the X Window architecture in general – is the low level of XKB specification. Policies or support for window managers – at ICCCM level at least – are not specified, so some usability features (eg. per-window group management, default group for new windows etc.) are not standardized: XKB leaves it up to the window manager/desktop environment to handle these issues.

Limitations such as the group number issue, and others, are not as significant as the fact that the classic XKB configuration scheme cannot manage all the configuration options properly, and does not fully utilize all 4 available groups.

Classic configuration repository and layout configuration (XFree 4.2.x and below)

The XKB configuration repository has 2 levels. At the low level, there are files describing the geometry of the keyboards, character codes, X keyboard symbols generated by keys, indicators and modifiers etc. The part which affects international users mostly is the directory of symbol descriptions. These “symbol files” specify the way keyboard symbols (keysyms) are translated to international characters (the standard Unicode-like encoding for X Window) – on a per-group basis. The group numbers are fixed in the symbol description files.

The high level in the repository deals with keyboard models, layouts, and options. The system of layouts (see Appendix A) is rule based – one rule per layout. Each layout combines one or more symbol descriptions. When loading a set of layouts (syntactically denoted by '+' characters) into the XKB configuration, a user gets a set of groups described in the corresponding XKB symbol description files – available for easy and fast switching between them. No combination of layouts allows the user to have more than 4 groups, due to the XKB protocol limitations mentioned above.

This way of merging layouts allows the user to use the groups in the same order as they are specified in original symbol files (see Appendix B). For example, if the group 'Armenian' (describing the mapping of the keysyms into Armenian characters) is put in the second position in the symbol file 'symbols/am' – you cannot move it to any other position in the resulting configuration, unless you use some other symbol file which puts the group 'Armenian' into the position you want. This affects the flexibility of the configuration scheme considerably. In an ideal world, where people can combine any group in any position, each national group should have 4 files which would put them in all positions 1-4. However that is not the case for XFree 4.2.x and earlier (and, to the best of author's knowledge, other X servers as well) - in most cases, a national symbols file contains one of 3 variants:

1. Local national group(s) in first position(s) – for most countries whose alphabets are Latin-based (eg. 'German').
2. A group for entering Latin characters (eg. 'US/ASCII') and a national group (eg. 'Russian').
3. A national group in the second position (eg.'Georgian') only – so it can be me merged with some 'Latin' one, which will take the first position.

Note that for the third variant above, the layout rule automatically added symbols from the file 'en_US' to the layout, so the first position is taken by "US/ASCII" group anyway.

Thanks to the inflexibility of this scheme, people effectively cannot use multiple non-Latin (actually, non 'US/ASCII') groups and layouts, even if they need only 2 or 3 of them. In order to do this, they have to manually create the symbol description files which would put necessary national groups into the appropriate positions.

Another consequence of this rigid infrastructure is the inability of the user to predict – without actually looking at the symbols files – which groups he or she is going to have by using and merging specific layouts. For example, there is no way that the user can predict that by using the layout 'ru', 2 groups will become available: 'US/ASCII' and 'Russian'. In contrast however, using the layout 'de' will give the user one group, 'German'. This makes the X keyboard configuration tools very cumbersome and illogical, and in the author's opinion there is no good X keyboard configuration tool for XFree 4.2.0: nothing compares to the flexibility of the manual editing of XF86Config file – assuming that the administrator knows what he is doing!

Another problem for the developers of the configuration tools is the lack of a holistic approach to the analysis of the configuration repository. The files describing the layout organization rules and their names ('rules/xfree86' and 'rules/xfree86.lst' in XFree) are in a non-standard format – X implementations do not provide a public parser API. These files also contain ambiguous and inexact data, so the X configuration tools either do not use these files or show the wrong information.

XKB configuration changes in XFree 4.3.0

In order to address the problems raised by the inflexible 'classic' configuration scheme, XFree 4.3.0 introduced several changes.

The most important change affected the way national groups from symbol files are merged (see Appendix C). The symbol descriptions were duplicated into separate subdirectory ('symbols/pc') in the source and distribution trees, and this new copy was modified to ensure that every symbol file contained only one national group in the first position. At the same time, each layout refers (by the rules) to only one symbol file. The merging of layouts put national groups into subsequent positions (for this type of merging, the character "," is used).

The most important consequence of this change is the ease of composition for 3-group and 4-group configurations. For example, in order to get 'American', 'Russian', 'German' and 'Greek' groups into one configuration, the user can merely merge the 'us', 'ru', 'de', and 'el' layouts: 'us,ru,de,el'. Another big improvement is the subsequent predictability of the resulting configuration for any combination of layouts. So the user, and the configuration tools, can make assumptions regarding layout names, symbols files and national groups in any direction.

In order to provide the full XKB configuration repository registry, the file 'rules/xfree86.xml' was introduced. This file contains exact, complete, and easy-to-parse information about available models, layouts and other XKB configuration parameters. This file can also be subject to localization (with the help of XML localization tools such as 'intltool').

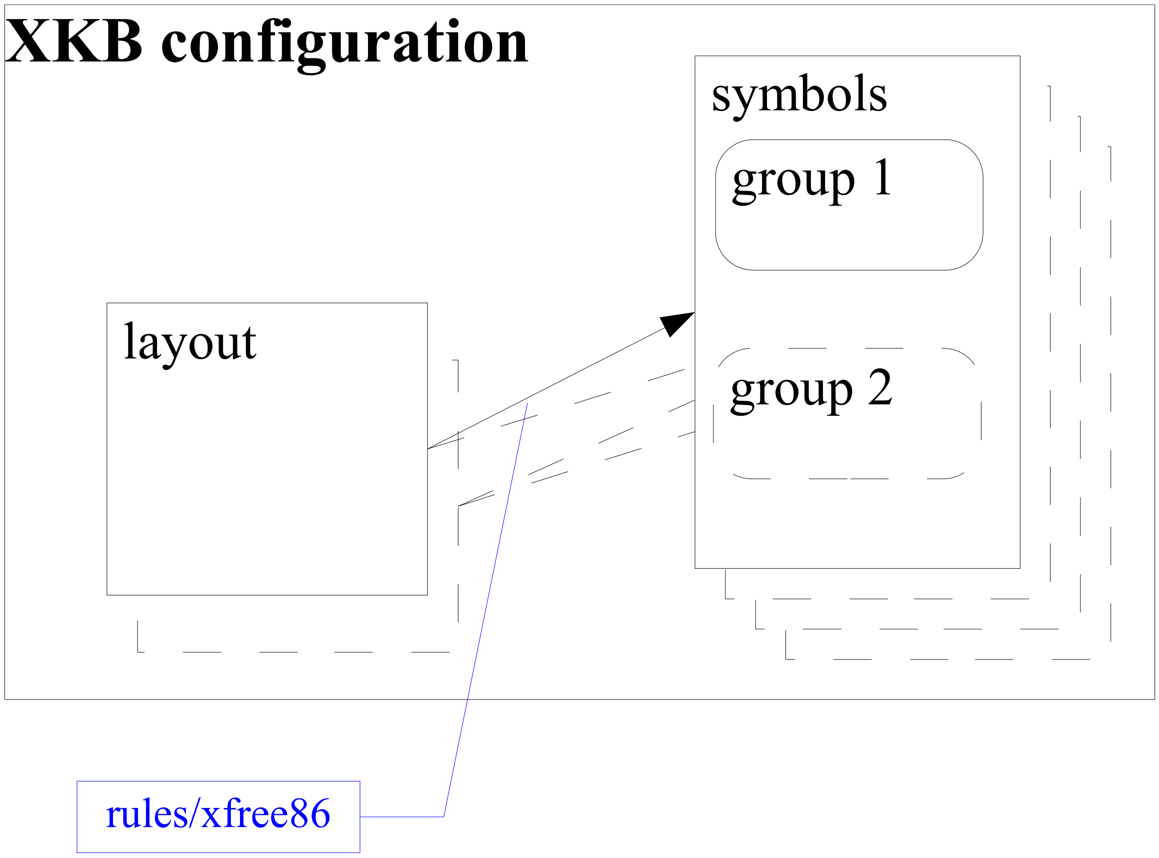
These changes have finally given developers the ability to create lightweight X keyboard configuration tools providing full and precise information about the available parameters and their possible values, allowing the user/administrator to predict the resulting configuration at the composition stage.

References

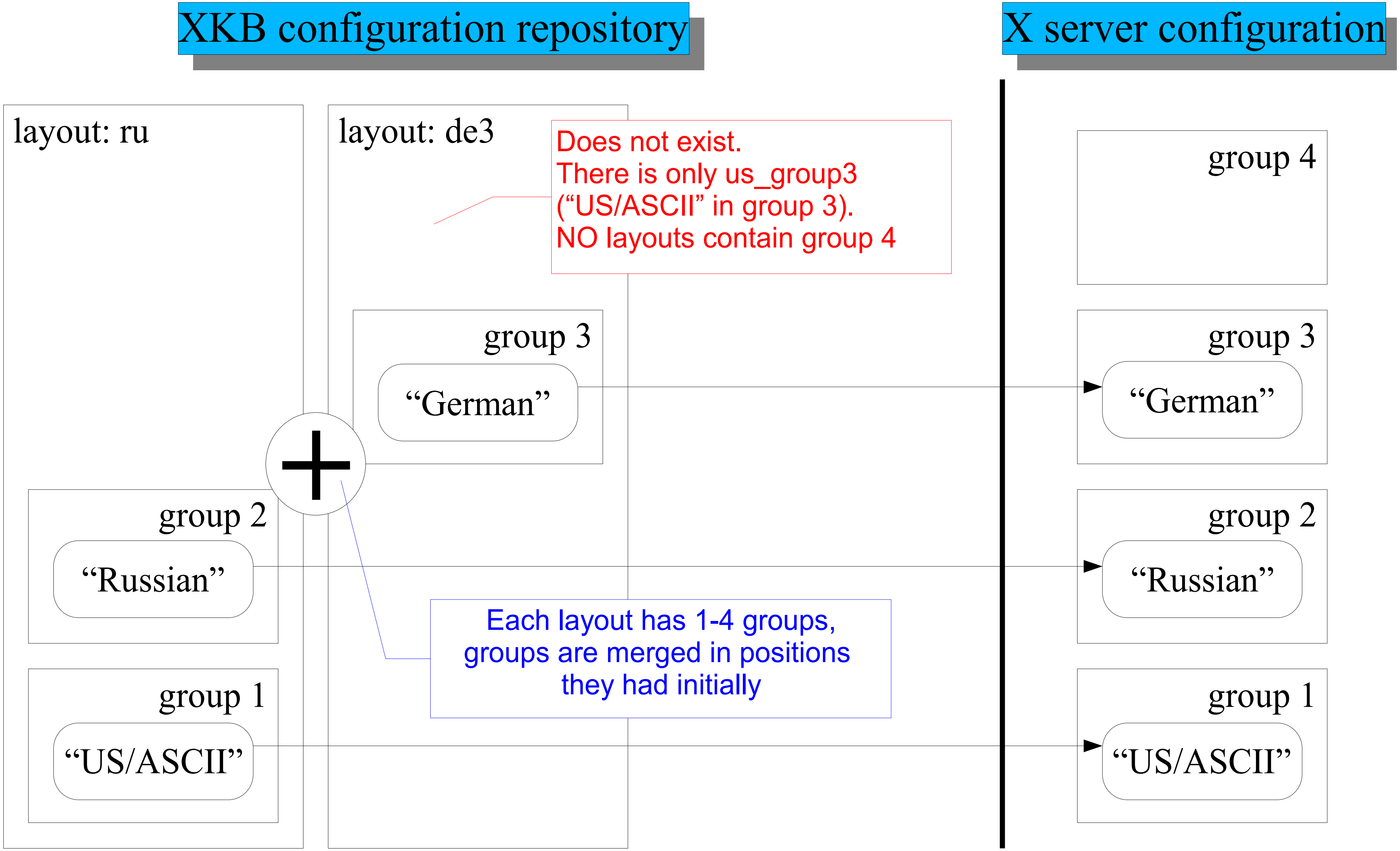
1. XKB Keyboard Extension: Protocol Specification

Appendices

Appendix A. Configuration, layouts, rules, symbols, groups



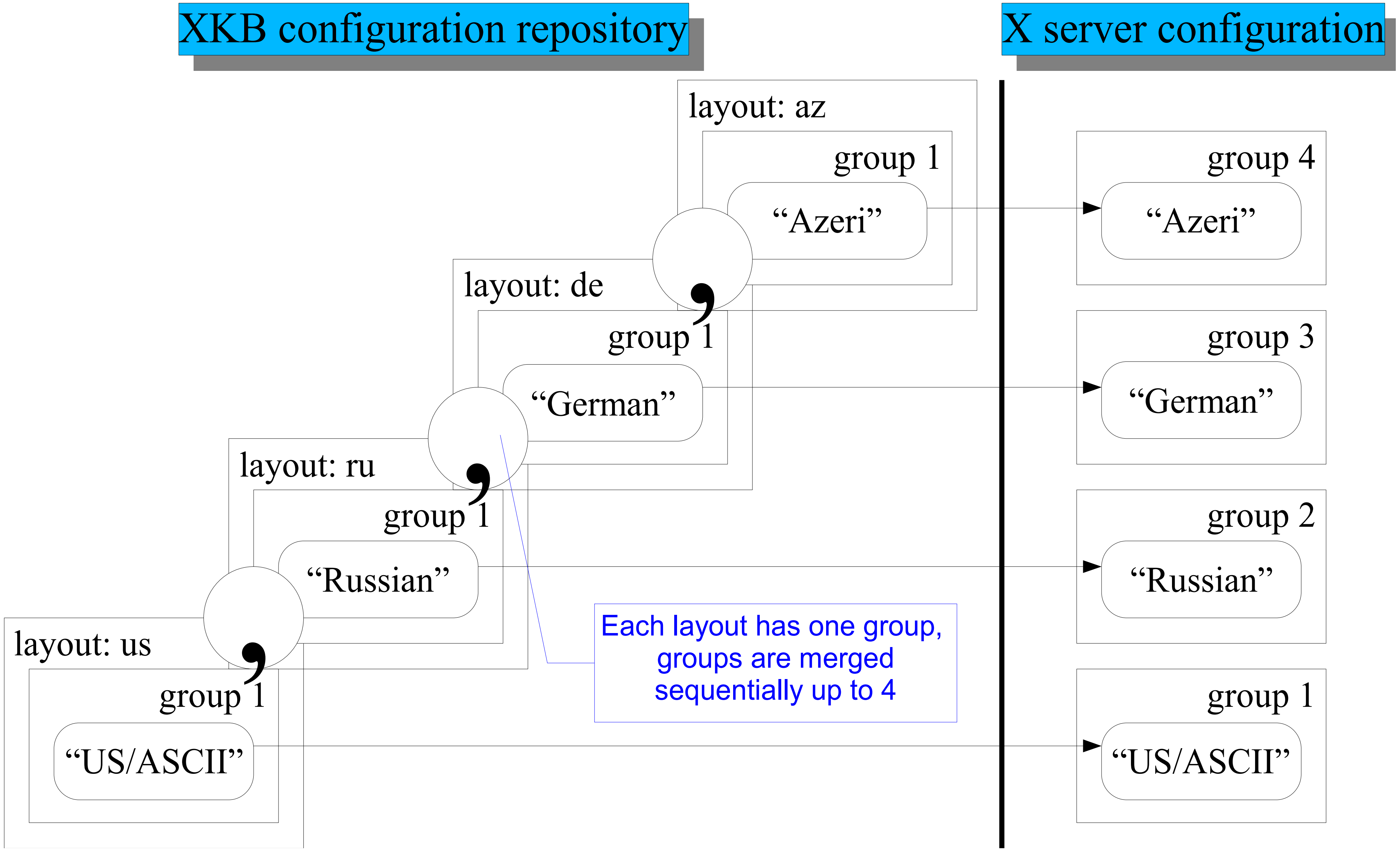
Appendix B. Classic XKB configuration scheme



Does not exist.
There is only us_group3
("US/ASCII" in group 3).
NO layouts contain group 4

Each layout has 1-4 groups,
groups are merged in positions
they had initially

Appendix C. New XKB configuration scheme


XKB configuration repository
X server configuration
layout: az
group 1
“Azeri”
layout: de
group 1
“German”
layout: ru
group 1
“Russian”
layout: us
group 1
“US/ASCII”
group 4
“Azeri”
group 3
“German”
group 2
“Russian”
group 1
“US/ASCII”
Each layout has one group,
groups are merged
sequentially up to 4